Probabilistic Pose Estimation of Deformable Linear Objects

Yujun Lai, James Poon, Gavin Paul, Haifeng Han and Takamitsu Matsubara

***Abstract*— This paper presents a probabilistic framework for online tracking of nodes along deformable linear objects. The proposed framework does not require an *a-priori* model; instead, a Bayesian Committee Machine, starting as a tabula rasa, accumulates knowledge over time. The key benefits of this approach are a lack of reliance upon extensive pre-training data, which can be difficult to obtain in sufficiently large quantities, and the ability for robust estimation of nodes subject to occlusion. Another benefit is that the uncertainties obtained during inference from the underlying Gaussian Processes can be beneficial towards subsequent handling tasks. Comparisons of the non-time series framework were conducted against conventional regression models to measure the efficacy of the proposed framework.**

I. Introduction

Deformable Linear Objects (DLOs) such as the one shown in Figure 1 are ubiquitous in everyday life with countless applications in both industrial and domestic settings. Over the past several years there has been an increase in the research conducted into the robotic manipulation of DLOs, in accordance with its perceived benefits towards the construction, manufacturing, and medical sectors.

Earlier investigations into DLO manipulation have been constrained in terms of perception such as non-occlusions in 2D space [9], and expert *a-priori* knowledge on the DLO topology [15]. The complexity of modeling DLOs has also been explored by [14] and [7], where visualizations of DLOs were utilized to predict their mechanical parameters. To overcome these simplifications, there is a need for a robust DLO tracker to facilitate reliable manipulation in real applications.

Previous works have explored object tracking in both 2D and 3D domains [10] [21], utilizing point registration based methodologies [20]. Contemporary works in occlusion handling have absorbed contextual information to create regions of interest [18] [25] for object segmentation. Probabilistic frameworks [19], filters [23], salient feature points [22], and background estimation [24] have also been able to detect, track, and handle occlusions. However, to the best of the authors' knowledge, there have been no works directly using the 3D geometric properties of the DLO to assist in occlusion handling.

Rather than treating occlusions as special-case events, a probabilistic approach is proposed here to robustly work with occlusions as they occur. To achieve this, pose estimation on nodes located along a DLO is performed using Gaussian Processes (GPs). This technique was chosen due to the uncertainties considered in GP inferences, which complements Bayesian updates of observed task knowledge. The flexibility of Bayesian updates and GPs have been exploited through previous works such as developing assistance models to mitigate cognitive deficiencies [6] and occupancy grid mapping in sub-optimal conditions [16] [17]. Other works have also utilised GPs specifically for human tracking through a variant – Gaussian Process Dynamical Models [11] [12]. As GPs scale poorly with a large number of training samples, a Bayesian Committee Machine (BCM) is used to incrementally collate data as it is made available.

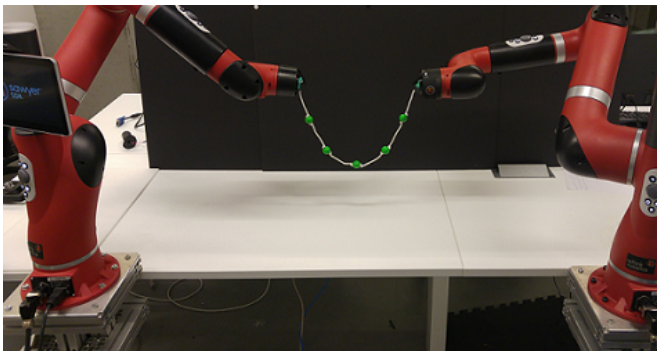


Fig. 1. Deformable Linear Object handled by two robotic manipulators.

The remainder of this paper is arranged as follows. Section II details the proposed framework, following a brief overview of GP and BCM. Section III documents the DLO experiments performed. Following a discussion of results in Section IV, Section V concludes with a summary of findings and potential directions for future investigations.

II. Methodology

In this section, a brief review of GPs for regression and BCM is followed by a description of the proposed DLO perception framework.

A. Gaussian Process Registration

Gaussian Processes [1] are one of the most promising techniques for building probabilistic regression models, given their simplicity and flexibility.

The non-linear function, $\boldsymbol{f}$ is first assumed as,

$$\boldsymbol{Y} = \boldsymbol{f}(\boldsymbol{X}) + \boldsymbol{\epsilon}, \tag{1}$$

where $\boldsymbol{Y}$ are training targets, $\boldsymbol{X}$ are training inputs, and $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}})$ is observation noise distributed normally with $\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}} = \text{diag}\left\{\sigma_1^2, \sigma_2^2, \ldots, \sigma_{d_y}^2\right\}$.

To learn such a relationship from data, a GP is fitted with the prior for each dimension of $\boldsymbol{y}$ as follows:

$$\boldsymbol{y}_a \sim \mathcal{GP}(0, K_a), \tag{2}$$

for $a = 1, 2, \ldots, d_{\boldsymbol{y}}$. The observation noise is $\epsilon_a \sim \mathcal{N}(0, \sigma_a^2)$.

The following squared exponential kernel function is considered for each element, k_a of K_a:

$$k_a(\boldsymbol{x}, \boldsymbol{x}') = \sigma_a^2 \exp\left(-\frac{1}{2l_a^2}||\boldsymbol{x} - \boldsymbol{x}'||^2\right), \tag{3}$$

where σ_a^2 is the variance of $\boldsymbol{f}_a$, and l_a is a scaling factor to normalize the distance between $\boldsymbol{x}$ and $\boldsymbol{x}'$. This kernel function was chosen for two reasons: it has been proven successful in capturing data correlations in a wide range of applications, and its simplicity makes it suitable for the analytical computation within the Bayesian estimation framework presented next.

As the maximum *a-posteriori* estimate of the kernel function hyper-parameter set $\boldsymbol{\theta}_a$ occurs where $p(\boldsymbol{Y}_a|\boldsymbol{X}, \boldsymbol{\theta}_a)$ is greatest, $\boldsymbol{\theta}_a$ is optimized as,

$$\log p(\boldsymbol{Y}_a|\boldsymbol{X}, \boldsymbol{\theta}_a) = -\frac{1}{2}\boldsymbol{Y}_a^{\mathrm{T}} K_a^{-1} \boldsymbol{Y}_a - \frac{1}{2}\log|K_a| - \frac{N}{2}\log 2\pi, \tag{4}$$

where N is the number of training data samples.

For inferring $\boldsymbol{y}_a^*$ at $\boldsymbol{x}^*$, the joint probability, $P(\boldsymbol{Y}_a, \boldsymbol{y}_a^{*\mathrm{T}})$ is modeled as a multivariate Gaussian distribution:

$$P\left(\begin{bmatrix}\boldsymbol{Y}_a \\ \boldsymbol{y}_a^*\end{bmatrix}\right) = \mathcal{N}\left(0, \begin{bmatrix}K_a & K_{a*}^{\mathrm{T}} \\ K_{a*} & K_{a**}\end{bmatrix}\right)$$

where $K_{a*} = [k_a(\boldsymbol{x}^*, \boldsymbol{X}_1), k_a(\boldsymbol{x}^*, \boldsymbol{X}_2), ..., k_a(\boldsymbol{x}^*, \boldsymbol{X}_N)]$ and $K_{a**} = k_a(\boldsymbol{x}^*, \boldsymbol{x}^*)$ respectively. The conditional distribution, $p(\boldsymbol{y}_a^*|\boldsymbol{Y}_a) = \mathcal{N}(K_{a*}K_a^{-1}\boldsymbol{Y}_a,\ K_{a**} - K_* K^{-1} K_{a*}^{\mathrm{T}})$ can be derived from this multivariate Gaussian.

B. The Bayesian Committee Machine

As GP inferences scale poorly at $\mathcal{O}(N^3)$ for N training samples, Bayesian Committee Machines (BCM) [2] alleviate this computational bottleneck by breaking down training data into M equally sized datasets, $D^{1,\ldots,M}$, building models for each D^i, where the number of data, $n \ll N$, and then combining their estimates. Note that here the index, a is dropped for clarity. In practice a BCM would be built for each dimension of $\boldsymbol{y}$. Given

$$P(y^*|D^{i-1}, D^i) \propto P(y^*)P(D^{i-1}|y^*)P(D^i|D^{i-1}, y^*), \tag{5}$$

and the approximation,

$$P(D^i|D^{i-1}, y^*) \approx P(D^i, y^*), \tag{6}$$

Bayes rule then yields,

$$P(y^*|D^{i-1}, D^i) = Q \times \frac{P(y^*|D^{i-1})P(y^*|D^i)}{P(y^*)}, \tag{7}$$

where Q is a constant. The resultant predictive distribution can then be approximated by:

$$\hat{P}(y^*|D^{1,\ldots,M}) = Q \times \frac{\prod_{i=1}^M P(y^*|D^i)}{P(y^*)^{M-1}}. \tag{8}$$

From M GP inferences, $\mathcal{N}(\mu_i, \sigma_i^2)$ at x^*, a prediction for the overall $\hat{y}^* = \mathcal{N}(\hat{\mu}^*, \hat{\sigma}^2)$ can be obtained:

$$\hat{\mu}^* = C^{-1} \times \sum_{i=1}^M \sigma_i^{-2}\mu_i, \tag{9}$$

where

$$C = \hat{\sigma}^{-2} = -(M-1)K_{**}^{-1} + \sum_{i=1}^M \sigma_i^{-2}. \tag{10}$$

C. Deformable Linear Object Tracking Framework

1) DLO Representation: The complexity of describing a DLO in Cartesian space makes it prohibitive for conventional GP modeling. Rather than mapping the incoming points in a latent space [13] to reduce the dimensionality, the geometric properties of DLOs are utilized instead.

Here the DLO is assumed to be hanging freely and subject to no other external forces besides gravity. Additionally, the DLO is taken as a series of nodes equally spaced along its length. As the DLO is inherently planar in this scenario, Figure 2 shows how the structure can be defined as a series of angular offsets, α between adjacent nodes.

2) Occluded α Estimation: The tracking objective is defined as $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, ..., \alpha_{n-1}\}^T$ for n physical nodes. Two Gaussian Process-Bayesian Committee Machines (GP-BCMs) are built, accumulating data over time to model α as a function of its neighbors, incorporating the ascending and descending uncertainty from both GP-BCMs; i.e. $\alpha_i = g(\alpha_{i-1}, \alpha_{i+1})$.

For clarity, here the index, i is dropped. From either of the BCM's estimates (Eqns. 9-10) of the first α in a sequence of several consecutive missing α values, the mean μ_f and its associated standard deviation, σ_f is obtained. This procedure is repeated on the other BCM to obtain μ_g and σ_g. An estimate of the next α's mean is simply $\hat{\mu_f} = f(\mu_f)$. Then from

$$(\mu_{lb}, \sigma_{lb}^2) = f(\mu_f - \sigma_f), \tag{11}$$

$$(\mu_{ub}, \sigma_{ub}^2) = f(\mu_f + \sigma_f), \tag{12}$$

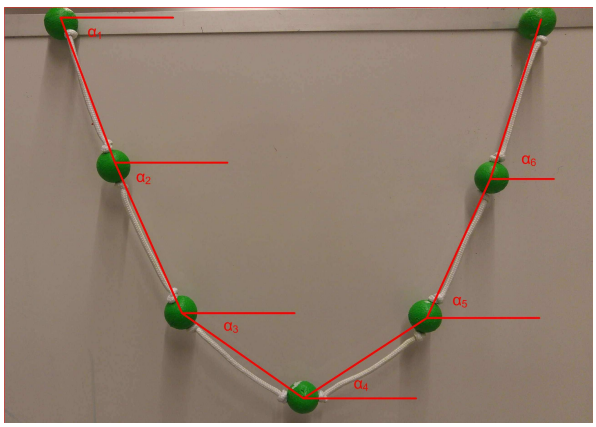


Fig. 2. Representation of α along the DLO.

we can approximate a bound for its ascending and descending uncertainties, σ_{asc} and σ_{desc} as,

$$\sigma_{asc} = (\mu_{ub} + \sigma_{ub}) - (\mu_{lb} - \sigma_{lb}), \tag{13}$$

$$\sigma_{desc} = (\mu_{lb} + \sigma_{lb}) - (\mu_{ub} - \sigma_{ub}), \tag{14}$$

$$\hat{\sigma}_f \approx \frac{1}{2} \times \max(|\sigma_{asc}|, |\sigma_{desc}|). \tag{15}$$

$\hat{\mu}_f$ and $\hat{\sigma}_f$ are then propagated in the same manner for the remaining α in the missing sequence. Although faster than the exact analytical solution in [8], this approximation assumes that $\hat{\sigma}_{f/g}$ remains reasonably constrained and that the latent BCM functions are not saddle points at $\hat{\mu}_{f/g}$.

Then α can be estimated from predictions, $(\hat{\mu}_f, \hat{\sigma}_f^2)$, $(\hat{\mu}_g, \hat{\sigma}_g^2)$ and weights, $w_f = \hat{\sigma}_f^{-2}$, $w_g = \hat{\sigma}_g^{-2}$ as,

$$\hat{\alpha} = \mathcal{N}\left(\frac{w_f\hat{\mu}_f + w_g\hat{\mu}_g}{w_f + w_g}, \frac{1}{w_f + w_g}\right). \tag{16}$$

III. Experiments

Experiments were conducted to evaluate the performance of the GP-BCM through two variants, and are compared against standalone GP models. For both variants, the subset size is explored to evaluate its effects on both runtime and estimation error.

A. Offline-trained GP-BCM

The offline-trained GP-BCM approach yields a BCM with M GPs of n training samples:

$$\alpha_{i,m} = g_m(\alpha_{i-1}, \alpha_{i+1}), \tag{17}$$

$$\alpha_{i,j} = g_j(\alpha_{i-1}, \alpha_{i+1}) \tag{18}$$

where $m, j < M$. The multiple underlying GPs in the BCM can mitigate the effects of noisy training data due to the weight-based estimations. However, we expect the noise filtering effect to eliminate any features present in the test data. The expected total training cost of this variant of GP-BCM reduces to $M \times n^3 < N^3$ where $n \ll N$.

B. Online GP-BCM

The online GP-BCM starts with no data samples, instead accumulating a subset over time. Once the accumulated data samples exceeds a maximum subset size, a GP model is trained utilizing the latest set of data. After more than GP has been trained, the framework will commence estimation of occluded α as they occur. Visible data is then used to build the subsequent GP model within the BCM, providing a continuous framework for estimation and training.

Noisy input data can be expected to be filtered out as more underlying GP models are built, lowering the sensitivity of the estimates to noise. Similar to the offline GP-BCM variant, as more underlying models are built over time, local features will be dissipated over time unless explicitly recovered.

C. Variance Propagation

As mentioned in Section II-C.2, the uncertainty of each α estimate can be approximated. This is then propagated for the estimation of each subsequent α in the missing sequence.

As the two GP-BCMs infer inwards from opposite ends, propagated uncertainties are softened through weights based on each GP-BCM's variance. Low variance estimates are prioritized and favored, providing improved confidence in the weighted estimates. These estimates are not included in the BCM's GP model generation, satisfying the assumption that variances are reasonably constrained (Section II-C.2).

D. Experiment Setup

One end of the DLO was manipulated by hand while the other end was fixed on a robot end-effector. When collecting data for experiments, all nodes are visible to serve as an occlusion-free ground truth for error calculations.

1) Occlusion of nodes: Occlusions are simulated by intentional omission of α measurements along a middle portion of the DLO, varying from no occlusions to 3 node occlusions. Such occlusions account for 25% of all data collected and simulate the natural occurrence of occlusions as illustrated by the sequence in Figure 3. For the GP-BCM experiments, the number of training samples for each GP was varied to evaluate the estimate accuracies. The number of occluded sample frames were normalized to the number of unique sample frames available.

2) Optimizing hyper-parameters for GP: The design of the optimization engine for GP hyper-parameters was also explored. Differential Evolution (DE) [5] was chosen as the optimization tool in this work. In contrast to gradient-based optimization tools such as Nelder-Mead Simplex [4] and Conjugate Gradient Method [3], DE does not require the optimization problem to be differentiable, allowing for robust handling of noisy inputs in continuous space.

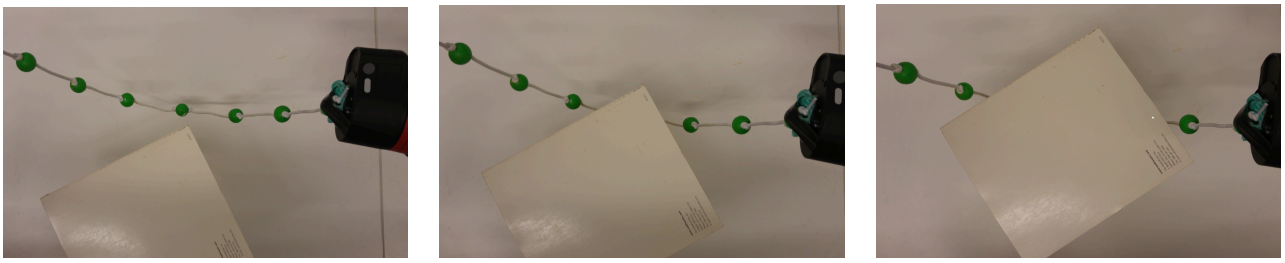
DE can be shown to have a negligible effect on computational efficiency (over time) between GP-BCM and GP as seen in Table I, with the number of iterations linearly proportional to training time. The chosen number of DE iterations and data input size can thus be arbitrarily chosen to fit the needs of the experiment. For the GP-BCM and standalone GP experiments, the number of DE iterations is set at 10 iterations.

TABLE I

Training Time for different Differential Evolution iteration numbers.

Based on 2005 sample size	Number of Differential Evolution iterations			
	5	10	15	20
Training Time	5.5356	8.2902	10.9902	13.8672
Line of Best Fit: $y = 2.7695x + 2.7471$				

As the input of the framework is based on angular representation (Section II-C.1), the possibility of multiple targets around an input is high. A major effect is the underfitting of inferences resulting in both erroneous and irrelevant estimates. Furthermore, training times would be expected



(a) (b) (c)

Fig. 3. Node sequences with: (a) no occlusion, (b) one node occlusion, and (c) three node occlusions in the dataset.

to increase due to the complexity of the hyper-parameter optimization, undermining the minimized training required for GP-BCM. To address this, training data was binned to yield a target mean and uncertainty for each discretized query.

The framework is assessed in terms of predictive accuracy and computational complexity:

1) Predictive Accuracy
   Two common error metrics used for estimation frameworks are Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). RMSE was chosen due to its symmetry and differentiability, complementing Gaussian distribution properties. Rather than weighting errors equally in MAE, RMSE is sensitive to outliers since its weighting is quadratic in nature. Similar to the non-linearity of Gaussian noise, RMSE provides a good metric to measure the accuracy of the proposed framework against the obtained ground truth.
2) Computational Complexity
   A common feature of GP-BCM variants and other frameworks is the training time involved for GP model generation. The efficiency of the proposed framework is defined as the unit GP model training time and is compared against conventional GPs. For online GP-BCM, the measurement includes the running time due to the online training of BCMs (noting the insignificance of inference computation cost).

The experiments were conducted based on 4 datasets obtained. The datasets vary in length from 401 to 743 unique sample frames. The size of the subset was explored ranging from 200 to 600 samples per GP model. Experiments were conducted on a machine with an Intel Core i7-7700@3.60GHz×8 CPU, 16GB RAM, and a Nvidia GeForce GTX 970 GPU.

TABLE II

TOTAL TIME (IN SECONDS) FOR CONVENTIONAL OFFLINE AND UNBINNED GP.

Dataset	1	2	3	4
No. samples	2005	2685	3715	3655
Training Time	300.49	530.52	1298.8	1223.1
Run Time	0.0394	0.0423	0.0474	0.0504
Efficiency (/sample)	0.1499	0.1976	0.3496	0.3347

IV. DISCUSSION

As the data is binned prior to GP model creation, the training time is capped to the number of bins. The effects of binning can be seen in Table II where the number of samples is non-linearly proportional to the training time required for a conventional GP.

From Table III, we see that GP-BCM variants have similar efficiency in terms of seconds per GP model produced (s/GP). We do note that online GP-BCM has a marginally lower efficiency when compared to offline GP-BCM. However, the training time required for online GP-BCM is, significantly lower. The online GP-BCM variant also allows continuous inference of α while training new GPs, thus providing scalability for testing.

TABLE III

TOTAL TIME FOR ALL GP-BCM VARIANTS IN SECONDS (S) FOR DATASET 1.

Test Variant		No. GPs	Training Time	Run Time	Efficiency (s/GP)
Offline GP-BCM	200	10	35.9875	0.0828	3.60703
	300	6	21.6174	0.0649	3.61372
	400	5	17.9483	0.0569	3.60104
	500	4	14.6243	0.0636	3.67198
	600	3	10.9066	0.0577	3.65477
Online GP-BCM	200	4	3.7616	11.0546	3.70405
	300	2	3.739	3.8381	3.78855
	400	2	3.7548	3.7545	3.75465
	500	1	3.7433	0.0623	3.8056
	600	1	3.7451	0.0591	3.8042

Given that there are 3 occluded nodes in the dataset (requiring 4 α inferences), an overall metric is required to evaluate the different variants within GP-BCM. The Sum of Squared Errors (SSE) is chosen to evaluate each variant's sensitivity to outliers due to the quadratic nature of the error metric, similar to RMSE. SSE's sensitivity to outliers forms a biased metric against outliers, making it optimal for the application since severe inference errors can derail subsequent frameworks manipulating a DLO.

From the results shown in Table IV, we can see that the online GP-BCM variant is comparable in performance to conventional GP. The GP sample sizes which produced the best result for online and offline GP-BCM were bolded, identifying a key characteristic of GP-BCM. We can see that the GP sample size affects the ability for the underlying GPs to provide accurate estimates as small sample sizes can

TABLE IV

ROOT MEAN SQUARE ERROR FROM EXPERIMENT RESULTS CONDUCTED WITH GP-BCM VARIANTS AND OFFLINE-TRAINED GP, AND SUM OF SQUARED ERRORS OF EACH VARIANT FOR ALL DATASETS.

RMSE (rads)	Online GP-BCM					Offline GP-BCM					Offline GP
	No. Samples					No. Samples					
	200	300	400	500	600	200	300	400	500	600	
Dataset 1 (401 sample frames)											
α_2	0.185	0.153	0.169	0.190	0.173	0.658	0.494	0.350	0.919	0.190	0.083
α_3	0.322	0.299	0.321	0.332	0.316	0.751	1.211	0.665	1.390	0.889	0.190
α_4	0.319	0.250	0.264	0.555	0.314	0.599	1.195	0.730	0.311	1.047	0.221
α_5	0.105	0.128	0.105	0.116	0.097	0.568	0.409	0.083	0.088	0.546	0.109
SSE	0.251	**0.192**	0.212	0.468	0.238	1.678	3.307	**1.105**	2.880	2.221	0.104
Dataset 2 (537 sample frames)											
α_2	0.010	0.018	0.010	0.010	0.009	2.742	2.982	3.548	1.680	0.868	0.024
α_3	0.063	0.087	0.045	0.042	0.044	1.917	2.725	2.890	1.498	0.703	0.046
α_4	0.303	0.362	0.034	0.044	0.023	0.380	0.431	0.383	0.334	0.528	0.064
α_5	3.011	0.994	0.241	0.260	0.172	0.208	0.145	0.131	0.150	0.248	0.034
SSE	9.161	1.127	0.061	0.071	**0.032**	11.38	16.52	21.10	5.199	**1.588**	0.008
Dataset 3 (743 sample frames)											
α_2	0.091	0.045	0.050	0.069	0.105	1.611	0.445	0.099	0.173	0.612	0.045
α_3	0.259	0.245	0.284	0.253	0.227	1.968	2.874	0.519	0.256	0.771	0.148
α_4	4.658	0.663	0.465	0.951	0.447	2.001	1.696	0.788	1.189	1.853	0.192
α_5	0.237	0.218	0.157	0.049	0.379	0.650	0.767	0.983	1.025	1.349	0.081
SSE	21.82	0.549	**0.324**	0.975	0.406	10.89	11.92	**1.867**	2.558	6.219	0.067
Dataset 4 (731 sample frames)											
α_2	0.201	0.201	0.197	0.180	0.200	0.493	0.646	0.213	0.218	0.265	0.153
α_3	0.181	0.284	0.202	0.176	0.220	0.574	0.581	0.367	0.294	0.696	0.138
α_4	0.647	1.279	0.449	0.177	0.153	0.390	0.949	0.345	0.527	1.039	0.165
α_5	0.179	0.270	0.178	0.187	0.190	0.563	0.633	0.411	0.455	0.597	0.152
SSE	0.523	1.829	0.312	**0.130**	0.147	1.041	2.056	**0.468**	0.619	1.989	0.093

potentially have a smaller range of training inputs, causing the GP model to be overconfident on incorrect estimates. This can be clearly seen in Dataset 2 for offline GP-BCM where α_2 and α_3 were consistently inaccurate when compared against α_4 and α_5.

The overall results of the framework were worse off in Datasets 3 and 4 due to the feature-rich nature of the dataset itself as seen in Figure 4. A noticeable dissipation of features within the dataset can be seen when the number of GP models increased (decreasing sample size will increase the number of underlying GP models). The preservation of local features can be seen when the number of GP models reaches a moderate value.In contrast to the feature-rich datasets, 3 and 4, the framework performed well when compared to conventional GPs with the feature-sparse datasets, 1 and 2.

In the results, we can see highly erroneous outputs to the dataset, specifically for the offline GP-BCM. Further investigation showed that the underlying model with the least variance lacked sufficient training data, skewing the weighted outputs. Every outlier result shared the common issue with one overconfident underlying GP model weighting heavily towards inaccurate regression inferences.

Although the mean and variance of the online GP-BCM is susceptible to errors through insufficient or biased training data, the capability to capture and integrate local features and trends with prior online data provides a distinct advantage over standalone GPs. In the context of machine learning, the uncertainty attached to the mean provides data-rich information for subsequent frameworks without the associated overheads. This is certainly the case for DLO manipulation frameworks which rely on expert or deterministic perception results.

V. CONCLUSION

A probabilistic framework is proposed for the online tracking of nodes along a deformable linear object using a Bayesian Committee Machine. By utilizing the computational efficiencies of BCM, an online tracking approach is realized.

To test the framework, experiments were conducted on inferring the angular displacement between concealed DLO nodes. An evaluative comparison against conventional offline Gaussian Process models and an offline BCM show promising results in inferring angular displacements of occluded nodes by considering their visible neighbors.

For future work, exploration of various ways to prioritize the most informative and relevant training data can be conducted to improve the accuracy of the framework. This can include inspecting the overconfident false outputs from the underlying GP models which skew the weighted outputs. The potential to incorporate non-planar movements of the end nodes exist, and can be explored to test the robustness of the framework.

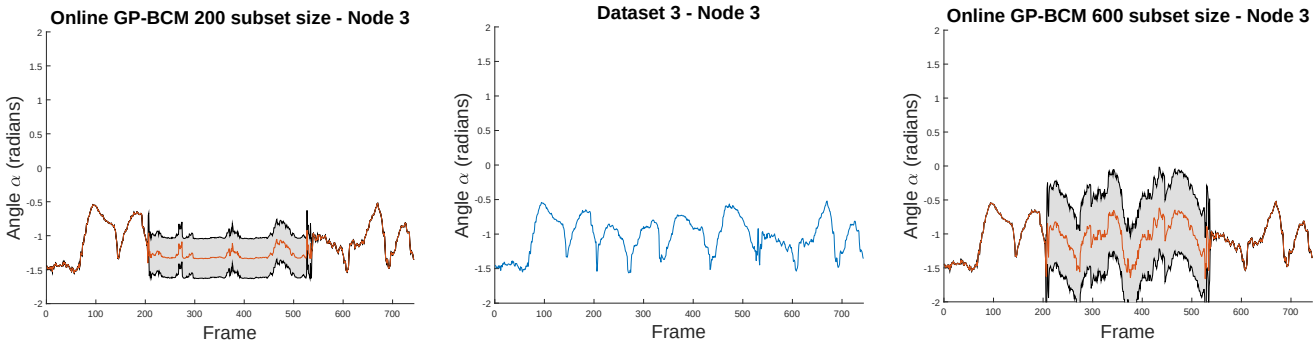


Fig. 4. The results of α_3 node from: (a) the 200 sample size online GP-BCM; (b) raw dataset plot; (c) the 600 sample size online GP-BCM. In (a), the framework filters the local features of (b) through the 4 (at the time of inference) underlying GP models built. In contrast, (c) shows the preservation of local features leading to better results for α_3 node for the dataset. The online GP-BCM's mean and its 95% confidence interval are indicated by the line plot and the shaded region respectively.

REFERENCES

[1] C. E. Rasmussen and C. K. I. Williams. *Gaussian Process for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, Cambridge, Massachusetts, 2005.

[2] V. Tresp. "A Bayesian Committee Machine". *Neural Computations*, vol. 12, pp. 2719–2741, Nov. 2000.

[3] K. G. Murty. "Quadratic Programming" in *Encyclopedia of Operations Research and Management Science*. Gass S.I., Fu M.C., Ed. Boston: Springer, 2013.

[4] J. A. Nedler, R. Mead. "A Simplex Method for Function Minimization". *The Computer Journal*, vol. 7, pp. 308–313, Jan. 1965.

[5] R. Storn and K. Price. "Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Space". *Journal of Global Optimization*, vol. 11, pp. 341–359, Nov. 1997.

[6] T. Matsubara, J. V. Miro, D. Tanaka, J. Poon, and K. Sugimoto. "Sequential intention estimation of a mobility aid user for intelligent navigational assistance". In *Proceedings of the 24th IEEE International Symposium on Robot and Human Interactive Communication*, 2015, pp. 444–449.

[7] N. Alvarez, K. Yamazaki, and T. Matsubara. "An Approach to Realistic Physical Simulation of Digitally Captured Deformable Linear Objects". In *IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*, 2016, pp. 135–140.

[8] J. Quiñonero-Candela, A. Girard, and C. E. Rasmussen. "Prediction at an Uncertain Input for Gaussian Processes and Relevance Vector Machines: Application to Multiple-Step Ahead Time-Series Forecasting". Technical University of Denmark, Denmark, 2003.

[9] H. Han, G. Paul and T. Matsubara. "Model-based Reinforcement Learning Approach for Deformable Linear Object Manipulation". In *IEEE Conference on Automation Science and Engineering (CASE)*, 2017, pp. 750–755.

[10] J. Schulman, A. Lee, J. Ho and P. Abbeel. "Tracking Deformable Objects with Point Clouds". In *IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 1130–1137.

[11] J. M. Wang, D. J. Fleet and A. Hertzmann. "Gaussian Process Dynamical Models for Human Motion". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 238–245, Feb. 2008.

[12] R. Urtasun, D. J. Fleet and P. Fua. "3D People Tracking with Gaussian Process Dynamical Models". In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 238–245, Jun. 2006.

[13] N. Lawrence and J. Quiñonero-Candela. "Local Distance Preservation in the GP-LVM through Back Constraints". In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, 2006, pp. 512–520.

[14] S, Javdani, S. Tandon, J, Tang, J, F. O'Brien and P, Abbeel. "Modeling and Perception of Deformable One-Dimensional Objects". In *IEEE Conference on Robotics and Automation (ICRA)*, 2011, pp. 1607–1614.

[15] M. Saha and P. Isto. "Manipulation Planning for Deformable Linear Objects". *IEEE Transaction on Robotics*, vol. 23, pp. 1141–1150, Dec. 2007.

[16] G. Paul, L. Liu, and D. Liu. "A Novel Approach to Steel Rivet Detection in Poorly Illuminated Steel Structural Environments". In *14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2016, pp. 1–7.

[17] G. Paul, D. Liu, N Kirchner, and G Dissanayake "An effective exploration approach to simultaneous mapping and surface materialtype identification of complex threedimensional environments". *Journal of Field Robotics*, vol. 26 (1112), pp. 915–933, Nov. 2009.

[18] J. Pan and B. Hu. "Robust Occlusion Handling in Object Tracking". In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007, pp. 1–8.

[19] N. Amezquite, R. Alquezar and F. Serratosa. "Dealing with Occlusion in a Probabilistic Object Tracking Method". In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2008, pp. 1–8.

[20] D. Hahnel, S. Thrun and W. Burgard. "An Extension of the ICP Algorithm for Modeling Nonrigid Objects with Mobile Robots". In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2003, pp. 915–920.

[21] A. Crivellaro, M. Rad, Y. Verdie, K. M. Yi, P. Fua and V. Lepetit. "Robust 3D Object Tracking from Monocular Images using Stable Parts". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, May 2017. [Online]. Available: https://doi.org/10.1109/TPAMI.2017.2708711[Feb. 26, 2018].

[22] M. M. Naushad Ali, M. Abdullah-Al-Wadud and S.-L. Lee. "Multiple object tracking with partial occlusion handling using salient feature points". *Information Sciences*, vol. 278, pp. 448–465, Sep. 2014.

[23] I. Iraei and K. Faez. "Object tracking with occlusion handling using mean shift, Kalman filter and Edge Histogram". In *IEEE 2nd International Conference on Pattern Recognition and Image Analysis (IPRIA)*, 2015, pp. 1–6.

[24] S. Zhao, S. Zhang and L. Zhang. "Towards Occlusion Handling: Object Tracking with Background Estimation". *IEEE Transactions on Cybernetics*, July 2017. [Online]. Available: https://doi.org/10.1109/TCYB.2017.2727138[Feb. 26,2018].

[25] D. T. Lin and Y. H. Chang. Occlusion Handling for Pedestrian Tracking using Partial Object Template-based Component Particle Filter. *IADIS International Journal on Computer Science and Information Systems*, vol. 8, pp. 40–50, 2013.