RKHSPose: Pseudo-keypoint RKHS Learning for Self-supervised 6DoF Pose Estimation

Yangzheng Wu, and Michael Greenspan

Introduction

- Problem:
  - Estimate the 6DoF pose of an object from an image
- Challenge:
  - Existing methods require labeled real data.
  - Synthetic data are accurate, and more efficiently generated.

Key Contributions

- Novel Approach: pseudo-keypoints for self-supervised learning of 6DoF pose estimation.
- RKHS Framework: Application of RKHS to learn a robust mapping from pseudo-keypoints to object pose.
- Efficiency: Achieve competitive performance without the need for real labeled data, reducing the need for expensive data annotation.

Methodology

- Pseudo-keypoints: points estimated using a synthetically pre-trained network by *overlaying the CAD model onto the real image (syn/real)*.
- Reproducing Kernel Hilbert Space (RKHS): RKHS to model and learn the relationships between these pseudo-keypoints and the object's pose in a self-supervised manner, by *mapping the feature spaces of the network with real image and syn/real image into the statistically comparable RKHS*.
- Self-Supervision: learns from the structure of the data itself, without labeled real datasets.

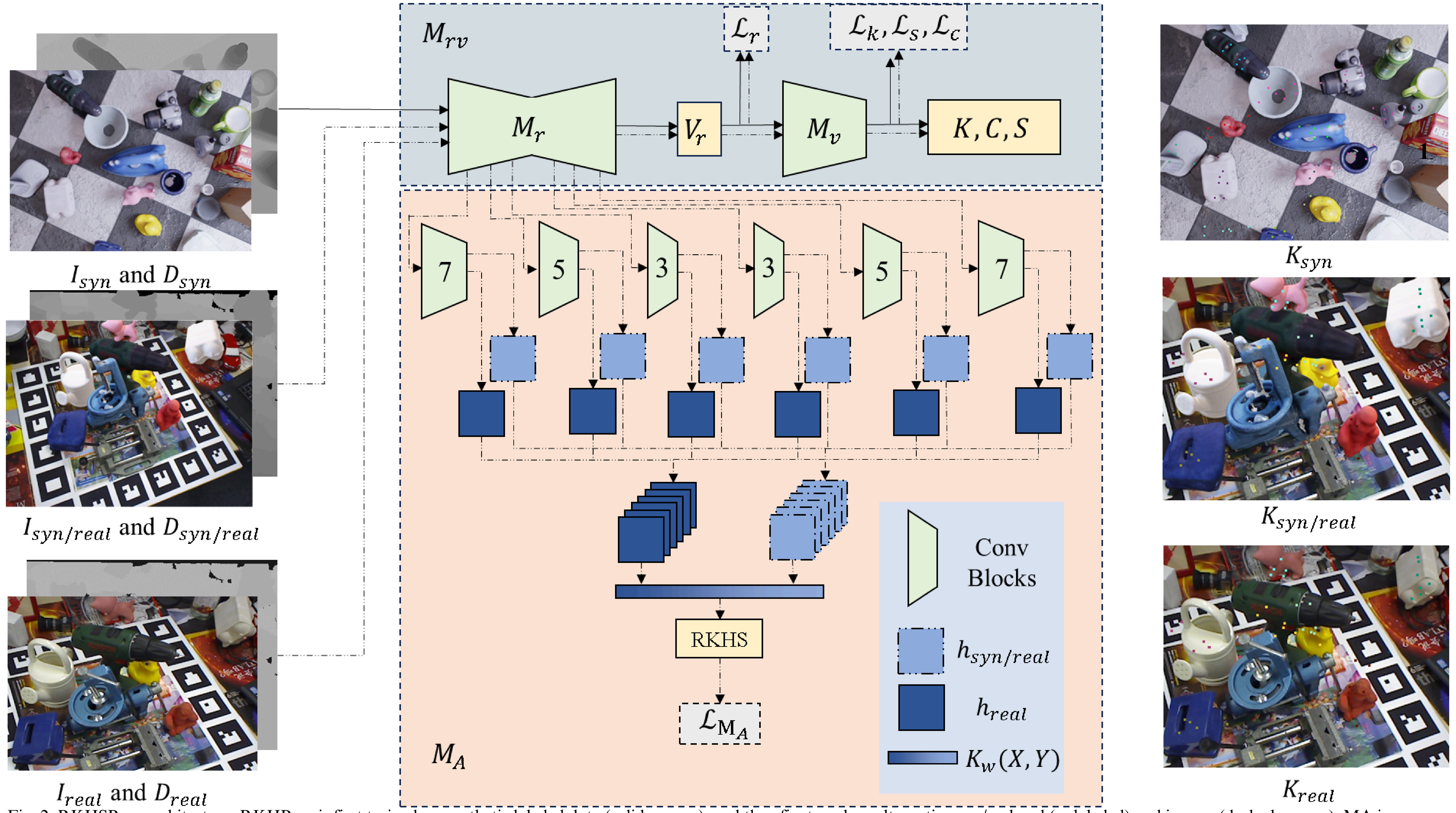

Fig. 2: RKHSPose architecture. RKHPose is first trained on synthetic labeled data (solid arrows), and then finetuned on alternating syn/real and (unlabeled) real images (dashed arrows). MA is measured by MMD in RKHS by densely mapping the intermediate features of Mr into high dimensional spaces with conv blocks. The distance is treated as LMA and back-propagated through MA and Mr.

Results

Performance: The method achieves SOTA results on standard benchmarks, demonstrating its effectiveness in real-world scenarios, as shown in Table 1.

Comparisons: It even outperforms traditional methods that rely on real data supervision, as shown in Table 2.

Table 1: Comparison with other methods. Accuracy of RKHSPose for LM and LMO is evaluated with ADD(S), and for YCB is evaluated with ADD(S) AUC. All 'Supervision: Syn + Self' methods use real images without real labels.

Method	Real data image	Real data label	LM ADD(S)	LMO ADD(S)	YCB ADD(S) AUC	YCB ADD-S AUC
Supervision: Syn (lower bound)						
AAE	✗	✗	31.4	-	-	-
MHP	✗	✗	38.8	-	-	-
GDR (TexPose version)	✗	✗	77.4	52.9	-	-
Self6D++	✗	✗	77.4	52.9	77.8	89.4
Self6D++ with D_{ref}	✗	✗	88.0	62.5	79.2	90.1
Ours	✗	✗	78.2	54.3	76.5	90.2
Ours+ICP	✗	✗	87.9	55.7	78.3	91.3
Supervision: Syn + Self						
Sock *et al.*	✓	✗	60.6	22.8	-	-
DSC	✓	✗	58.6	24.8	-	-
Self6D	✓	✗	58.9	32.1	-	-
SMOC-Net	✓	✗	91.3	63.3	-	-
Self6D++	✓	✗	88.5	64.7	80.0	91.4
TexPose	✓	✗	91.7	66.7	-	-
Ours	✓	✗	95.8	68.6	82.8	92.4
Ours+ICP	✓	✗	**95.9**	**68.7**	**83.0**	**92.6**
Supervision: Syn + Real GT (upper bound)						
SO-Pose	✓	✓	96.0	62.3	83.9	90.9
Self6D++	✓	✓	91.0	74.4	82.6	90.7
Ours	✓	✓	96.7	70.8	85.4	92.2
Ours+ICP	✓	✓	**96.8**	**71.3**	**85.6**	**92.4**

Table 2: Comparison with fully supervised methods. RKHSPose results on TLESS (−1.8), TUDL (−0.4), ITODD (−4.6) and HB (+0.1) compares to SOTA methods with full supervision of real GT labels. Methods annotated with * use the detection results from other detection methods.

Method	real label	LM	LMO	TLESS	TUDL	ITODD	HB	YCB
SurfEmb* [30]	✓	-	76.0	82.8	85.4	65.9	86.6	79.9
RCVPose3D [82]	✓	-	72.9	70.8	**96.6**	**73.3**	86.3	84.3
RADet [87]+PFA* [39]	✓	-	**79.7**	85.0	96.0	67.6	86.9	88.8
ZebraPose [70]	✓	-	78.0	**86.2**	95.6	65.4	92.1	**89.9**
Ours	✗	95.7	68.2	85.5	96.2	68.6	92.2	83.6
Ours+ICP	✗	95.8	68.4	85.6	96.2	68.7	**92.3**	83.8

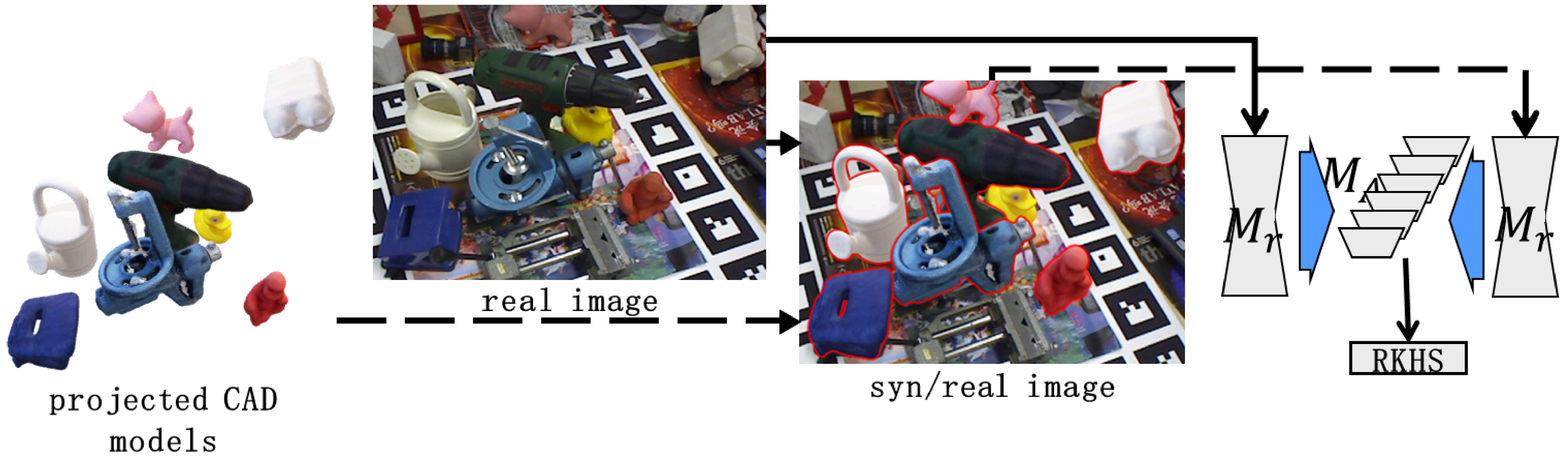

Fig. 1: RKHSPose adapts the network pretrained on synthetic data to real test scenes (left), by comparing network feature spaces with real image inputs (solid arrows), against those with syn/real image (right) inputs (dashed arrows). Mr regresses radial quantities, MA is the Adapter network, and RKHS maps features into a higher dimensional space.

Ablation Studies:

- Adapter with different kernels
- Dense Vs. Sparse Adapter
- Syn/Real Synchronized Training
- Adapter Kernels and Metrics
- Influence of Real GT Labels

Table 6: *AR* of different kernels on LM and five BOP core datasets.

Kernel	w	AR_{VSD}	AR_{MSSD}	AR_{MSPD}	AR
Linear	✗	71.6	70.8	70.6	71.0
	✓	84.9	84.1	84.3	**84.4**
RBF	✗	73.4	72.9	73.2	73.2
	✓	82.5	81.3	81.5	81.6

Table 3: *AR* of different adapters on LM and five BOP core datasets.

Adapter	AR_{VSD}	AR_{MSSD}	AR_{MSPD}	AR
M_A^s	78.1	77.8	77.8	77.9
M_A	84.9	84.1	84.3	**84.4**

Table 4: *AR* of different training strategies on LM and five BOP core datasets.

Training Type	AR_{VSD}	AR_{MSSD}	AR_{MSPD}	AR
Mixed	84.9	84.1	84.3	**84.4**
Sequantial	82.3	81.7	81.7	81.9

Table 5: *AR* of different metrics on LM and five BOP core datasets.

Metric	AR_{VSD}	AR_{MSSD}	AR_{MSPD}	AR
MMD	84.9	84.1	84.3	**84.4**
KL Div	78.0	77.8	78.0	77.9
Wass	80.9	80.6	80.9	80.8

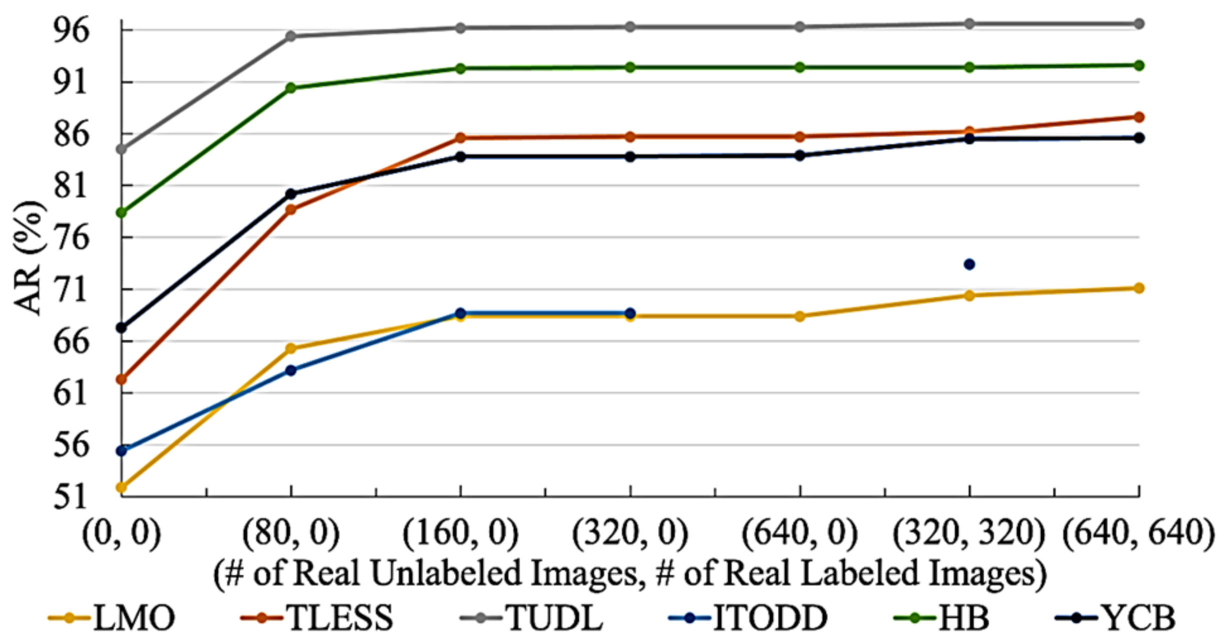

Fig. 3: Impact of # of real images with/without GT labels used during training. All datasets are evaluated by the BOP AR metric. We conduct experiments from 0 to 640 real images on all datasets, except ITODD which contained only 357 real images.

Conclusion

Impact: This work provides a significant step towards more autonomous and scalable 6DoF pose estimation.

Future Work: The potential for extending this approach to more complex scenes and objects, as well as integrating it into real-time systems.

Sponsored by Bluewrist Inc. Canada and NSERC.