Robust Monocular Visual-Inertial Depth Completion for Embedded Systems

Nathaniel Merrill*, Patrick Geneva*, and Guoquan Huang
University of Delaware, USA

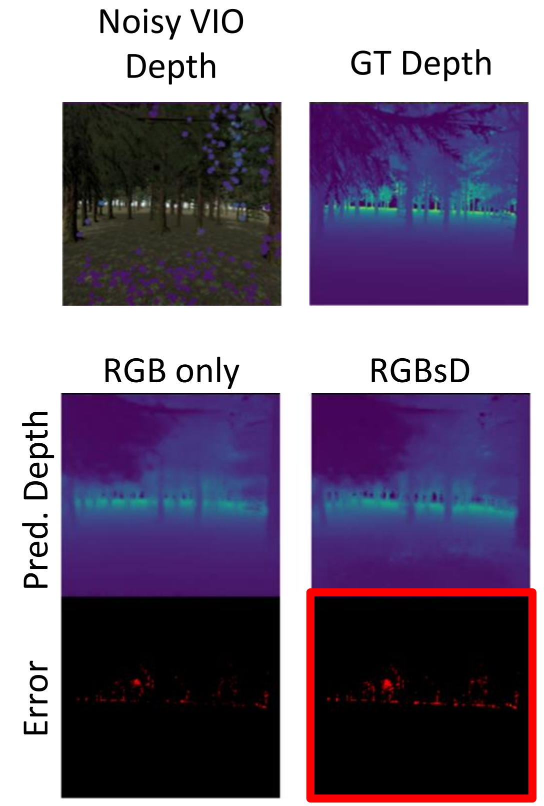
Introduction

- Goal: Provide real-time accuracy **dense depth** and **6DoF pose** estimates on embedded systems for planning and control
- Leverage sparse VIO depth for accurate depth completion with only **a camera and IMU**
- Contributions:
  - **Real-time** visual-inertial estimation and depth completion on embedded devices
  - Investigation of depth completion RGBsD **sensitivities** and robust training schemes
  - Demonstrate evaluation on embedded devices

Example navigation through forest environment at high speed

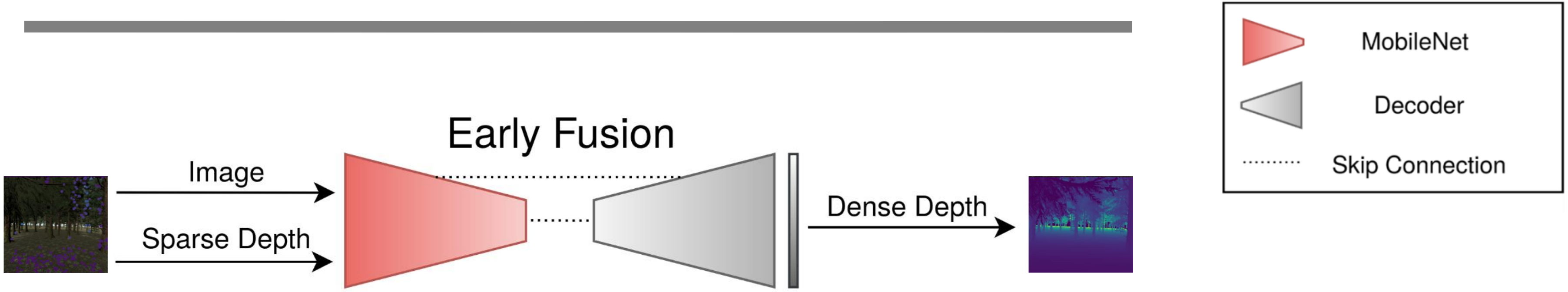
Depth Completion

- Single-view depth (RGBsD) enables real-time completion and recovery of fully dense depth maps
- Sparse depth allows RGBsD to have **improved accuracy** compared to RGB only
- Key Observations:
  - Existing methods fail under **noisy** VIO sparse depths, negating benefits of leveraging sparse depth



Larger error then RGB with typical RGBsD network!

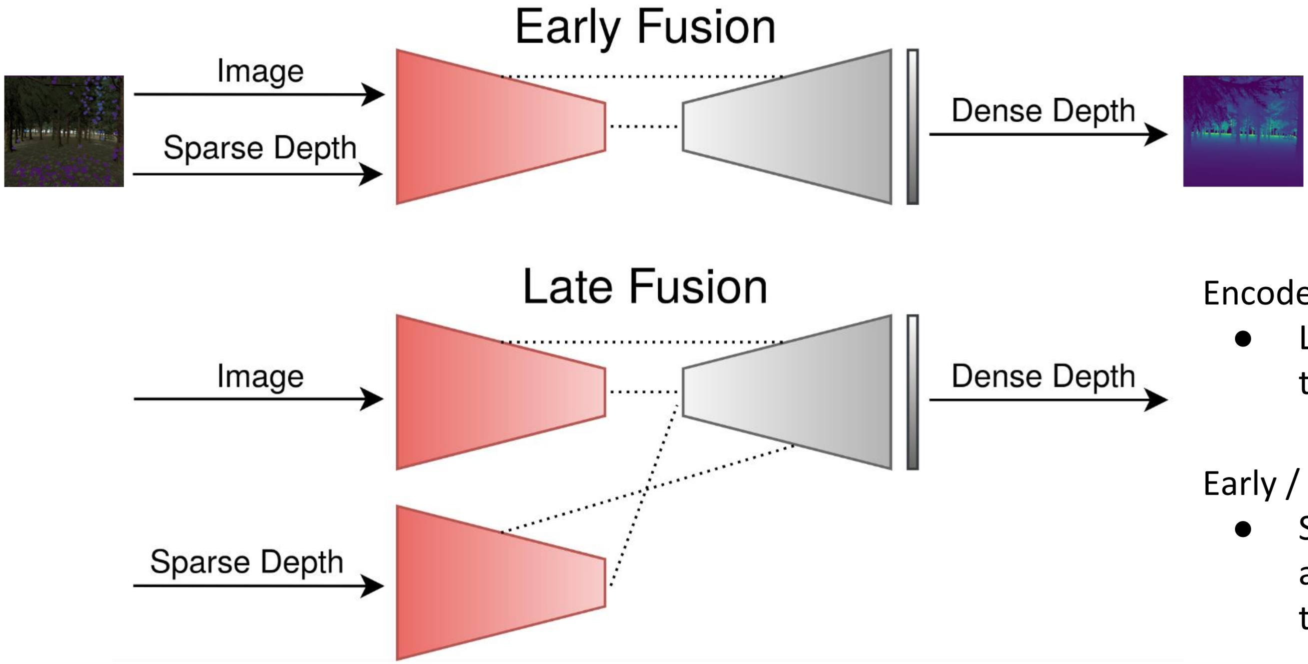
Network Architecture

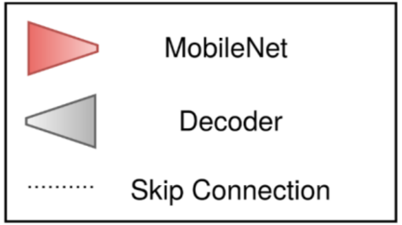


Encoder/Decoder FastDepth [Wofk2019]

- Lightweight architecture geared towards embedded devices

Network Architecture





Encoder/Decoder FastDepth [Wofk2019]

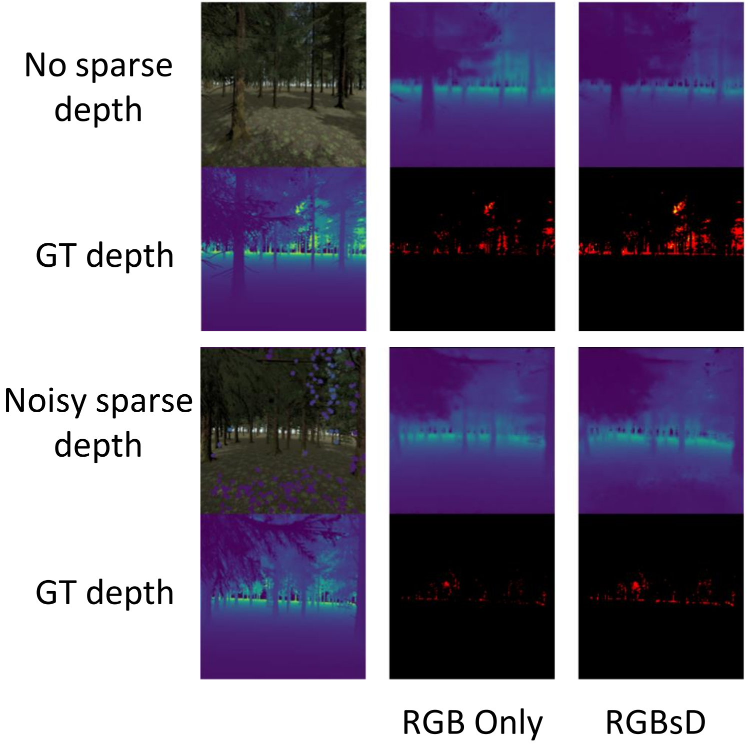
- Lightweight architecture geared towards embedded devices

Early / Late Fusion:

- Sparse depth either appended into a common encoder or passed through separate encoder

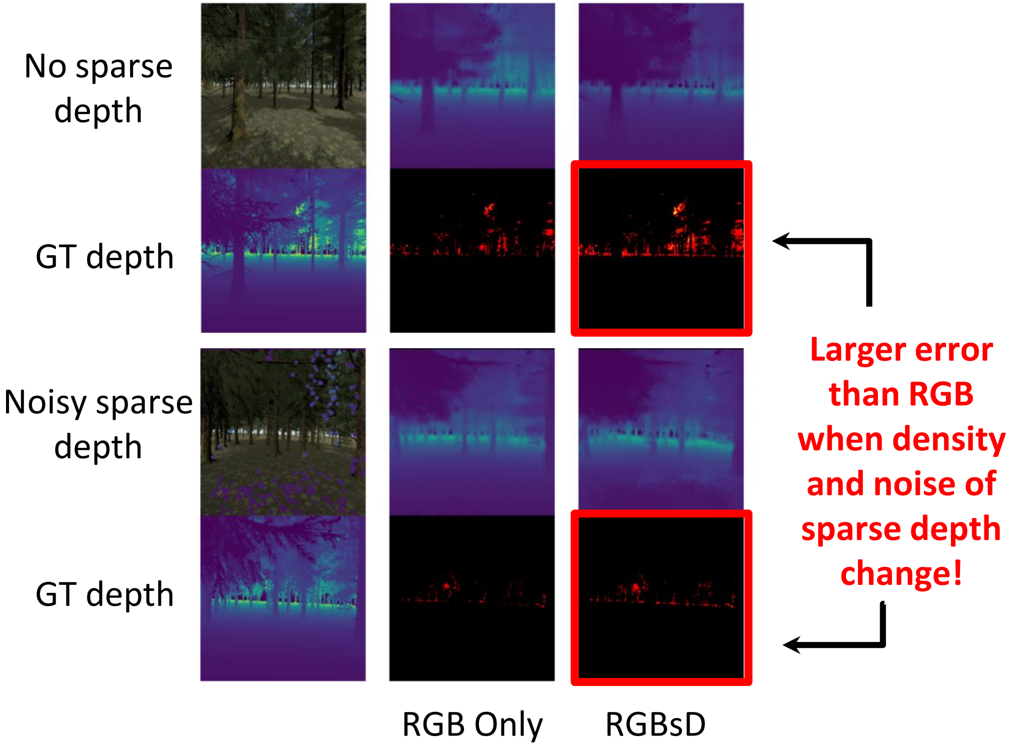
Typical Sparse Depth Completion

- Most existing RGBsD networks are trained with **uniformly sampling** GT depth
- VIO depths are: **salient features, noisy, with varying density**
  - RGBsD worse than RGB!



Typical Sparse Depth Completion

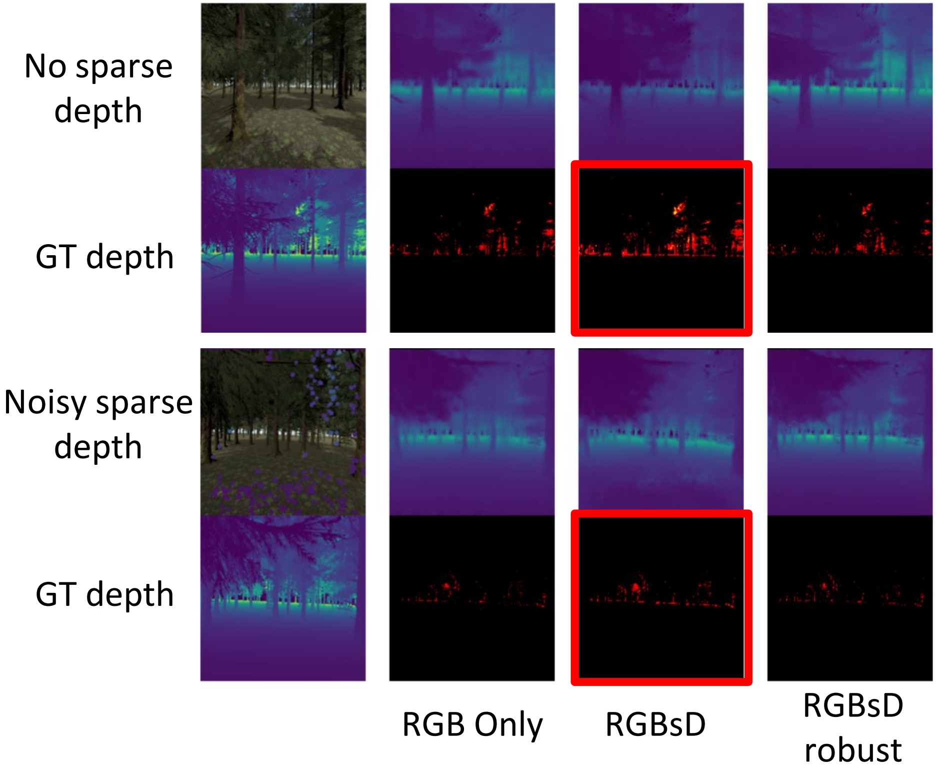
- Most existing RGBsD networks are trained with **uniformly sampling** GT depth
- VIO depths are: **salient features, noisy, with varying density**
  - RGBsD worse than RGB!



Typical Sparse Depth Completion

- Most existing RGBsD networks are trained with **uniformly sampling** GT depth
- VIO depths are: **salient features, noisy, with varying density**
  - RGBsD worse than RGB!

The proposed **robust training** and **initialization** scheme ensures depth accuracy are the same or better then RGB



Robust Training Scheme

Training Scheme	Model	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$	RMSE $\downarrow$	MAE $\downarrow$	$\log_{10} \downarrow$
baseline	VIO (sparse)	0.651	0.790	0.848	6.780	3.179	0.126
	RGB	0.604	0.841	0.913	5.838	3.332	0.125

Baseline accuracy of sparse points and RGB only

Robust Training Scheme

Training Scheme	Model	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$	RMSE $\downarrow$	MAE $\downarrow$	$\log_{10} \downarrow$
baseline	VIO (sparse)	0.651	0.790	0.848	6.780	3.179	0.126
	RGB	0.604	0.841	0.913	5.838	3.332	0.125
typical	RGBsD	0.431	0.674	0.802	8.736	5.083	0.188
	RGBsD-late	0.539	0.748	0.830	6.390	3.434	0.147

RGBsD Color Legend

- Better than RGB
- < 10% worse than RGB
- ≥ 10% worse than RGB

Poor prediction accuracy using traditional training methods

Robust Training Scheme

Training Scheme	Model	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$	RMSE $\downarrow$	MAE $\downarrow$	$\log_{10} \downarrow$
baseline	VIO (sparse)	0.651	0.790	0.848	6.780	3.179	0.126
	RGB	0.604	0.841	0.913	5.838	3.332	0.125
typical	RGBsD	0.431	0.674	0.802	8.736	5.083	0.188
	RGBsD-late	0.539	0.748	0.830	6.390	3.434	0.147
sim VIO	RGBsD	0.445	0.774	0.874	7.058	4.120	0.155
	RGBsD-late	0.570	0.774	0.867	7.287	3.998	0.141

RGBsD Color Legend

- Better than RGB
- < 10% worse than RGB
- ≥ 10% worse than RGB

Even if we perform data augmentation to robustly train, still worst performance!

Robust Training Scheme

RGBsD Color Legend

- Better than RGB
- < 10% worse than RGB
- ≥ 10% worse than RGB

Training Scheme	Model	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$	RMSE $\downarrow$	MAE $\downarrow$	$\log_{10} \downarrow$
baseline	VIO (sparse)	0.651	0.790	0.848	6.780	3.179	0.126
	RGB	0.604	0.841	0.913	5.838	3.332	0.125
typical	RGBsD	0.431	0.674	0.802	8.736	5.083	0.188
	RGBsD-late	0.539	0.748	0.830	6.390	3.434	0.147
sim VIO	RGBsD	0.445	0.774	0.874	7.058	4.120	0.155
	RGBsD-late	0.570	0.774	0.867	7.287	3.998	0.141
sim VIO w/ RGB init	RGBsD	0.582	**0.852**	**0.919**	**5.652**	3.318	0.127
	RGBsD-late	**0.658**	0.849	0.915	5.742	**3.211**	**0.122**

Proposed Solution:

- Pretrain the network with RGB only depth
- Train with *simulated* VIO depths to robustify
- Ensures performance does not drop worst then RGB only!

Comparison to Sparse-to-Dense [Ma 2018]

- Sparse-to-Dense (S2D) network
  - Based on more powerful ResNet model
  - Trained with uniform noise-free sparse depth

Testing on NYUv2 with Sampled Sparse Depth

Test	**Model**	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$	**RMSE** $\downarrow$	**MAE** $\downarrow$	**abs. rel.** $\downarrow$	$\log_{10} \downarrow$
-	FastDepth	0.904	0.971	0.989	0.433	0.267	0.092	0.042
uniform	S2D	**0.981**	**0.995**	**0.998**	**0.194**	**0.100**	**0.036**	**0.016**
	RGBsD	0.956	0.991	0.997	0.281	0.173	0.063	0.028
	RGBsD-late	0.963	0.991	0.997	0.270	0.160	0.057	0.025

Comparison to Sparse-to-Dense [Ma 2018]

- Sparse-to-Dense (S2D) network
  - Based on more powerful ResNet model
  - Trained with uniform noise-free sparse depth

Testing on NYUv2 with Sampled Sparse Depth

Test	Model	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$	RMSE $\downarrow$	MAE $\downarrow$	abs. rel. $\downarrow$	$\log_{10} \downarrow$
-	FastDepth	0.904	0.971	0.989	0.433	0.267	0.092	0.042
uniform	S2D	**0.981**	**0.995**	**0.998**	**0.194**	**0.100**	**0.036**	**0.016**
	RGBsD	0.956	0.991	0.997	0.281	0.173	0.063	0.028
	RGBsD-late	0.963	0.991	0.997	0.270	0.160	0.057	0.025
corners	S2D	0.626	0.689	0.730	1.247	0.777	0.267	0.199
	RGBsD	0.948	0.990	0.997	0.301	0.194	0.072	0.032
	RGBsD-late	**0.955**	**0.990**	**0.997**	**0.286**	**0.176**	**0.064**	**0.028**

Comparison to Sparse-to-Dense [Ma 2018]

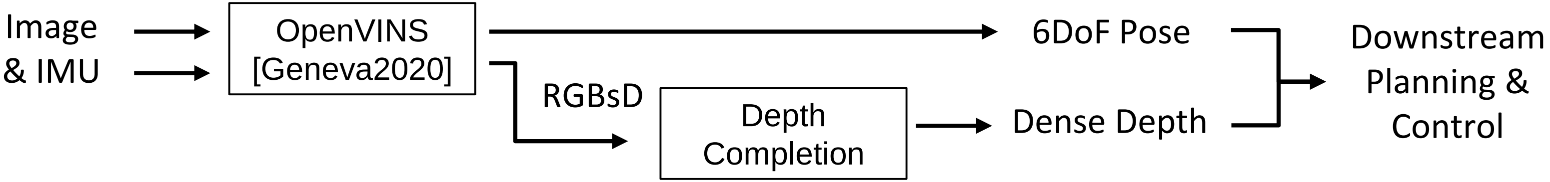
- Sparse-to-Dense (S2D) network
  - Based on more powerful ResNet model
  - Trained with uniform noise-free sparse depth

Testing on NYUv2 with Sampled Sparse Depth

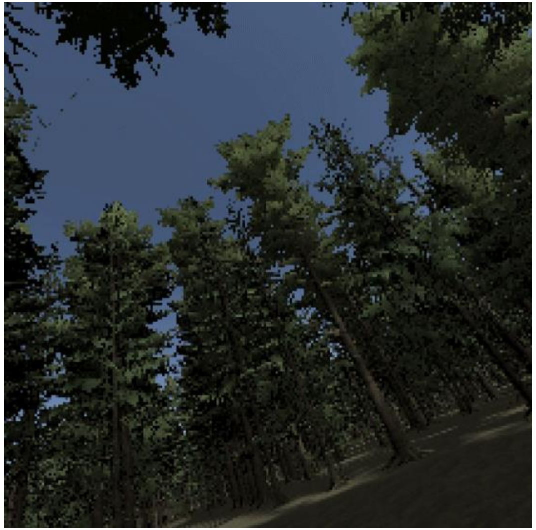
Test	Model	δ_1 ↑	δ_2 ↑	δ_3 ↑	RMSE ↓	MAE ↓	abs. rel. ↓	$\log_{10}$ ↓
-	FastDepth	0.904	0.971	0.989	0.433	0.267	0.092	0.042
uniform	S2D	**0.981**	**0.995**	**0.998**	**0.194**	**0.100**	**0.036**	**0.016**
	RGBsD	0.956	0.991	0.997	0.281	0.173	0.063	0.028
	RGBsD-late	0.963	0.991	0.997	0.270	0.160	0.057	0.025
corners	S2D	0.626	0.689	0.730	1.247	0.777	0.267	0.199
	RGBsD	0.948	0.990	0.997	0.301	0.194	0.072	0.032
	RGBsD-late	**0.955**	**0.990**	**0.997**	**0.286**	**0.176**	**0.064**	**0.028**
noisy corners	S2D	0.480	0.613	0.680	1.371	0.980	0.350	0.241
	RGBsD	0.942	0.988	0.996	0.316	0.203	0.075	0.032
	RGBsD-late	**0.950**	**0.989**	**0.996**	**0.298**	**0.186**	**0.069**	**0.030**

S2D significantly worse than FastDepth (RGB-only) with **out-of-distribution** sparse depths!

Challenging Forest Application



- Complete system demonstrated on challenging simulated forest dataset
- Trained using proposed method and highly variable viewpoints
- Challenges:
  - **Large depth range** due to gaps in trees
  - **High detail level** due to vegetation



Deployment to Embedded Platforms

- Key OpenVINS modifications:
  - Limit features for update
  - **Out-of-state features** for sparse depth-map generation
- NVIDIA Jetson devices allow depth completion GPU acceleration
- Leveraged Apache TVM **autotune optimization** to further tune network prediction speed (x2 speedup)

Nano

- 128-core Maxwell
- Quad-core ARM A57 @ 1.43 GHz
- 4 GB 64-bit LPDDR4
- Max 10W power

TX2

- 256-core Pascal
- Dual-core Denver 2 64-bit CPU & quad-core ARM A57 complex
- 4 GB 128-bit LPDDR4
- Max 15W power

[redacted]

Timing Results

- Both depth prediction and OpenVINS on NVIDIA Nano and TX2 are able to be **real-time**
- OpenVINS single-threaded performance split between EKF update and feature tracking
- Complementary resource usage of **single CPU thread and GPU** leaving compute for planning & control

Network

Platform	Model	No-Optimization			TVM-Optimization		
		t_{pred}	σ	Hz	t_{pred}	σ	Hz
Nano (GPU)	RGB	24.75	0.69	40	14.90	0.79	67
	RGBsD	24.91	0.58	40	17.07	1.03	58
	RGBsD-late	42.99	0.46	23	70.66	0.34	14
TX2 (GPU)	RGB	10.79	0.46	92	6.87	0.70	145
	RGBsD	10.78	0.60	92	7.09	0.71	141
	RGBsD-late	18.36	0.47	54	11.34	0.90	88

Minimal overhead from including sparse depth!

OpenVINS

Platform	t_{track}	t_{total}	σ_{total}
Nano	0.0192	0.0359	0.0115
TX2	0.0128	0.0280	0.0088
Desktop	0.0085	0.0183	0.0044

OpenVINS on the embedded devices at ~30Hz

Conclusion

- Showed that noisy VIO depths can **significantly hurt** depth completion
- Proposed **robust** training strategy
  - Initialize to RGB-only weights
  - Train with **noisy** sampled corner features with imperfect depths
- Demonstrated real-time VIO depth completion on **embedded** devices

Nathaniel Merrill
nmerrill@udel.edu