DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras

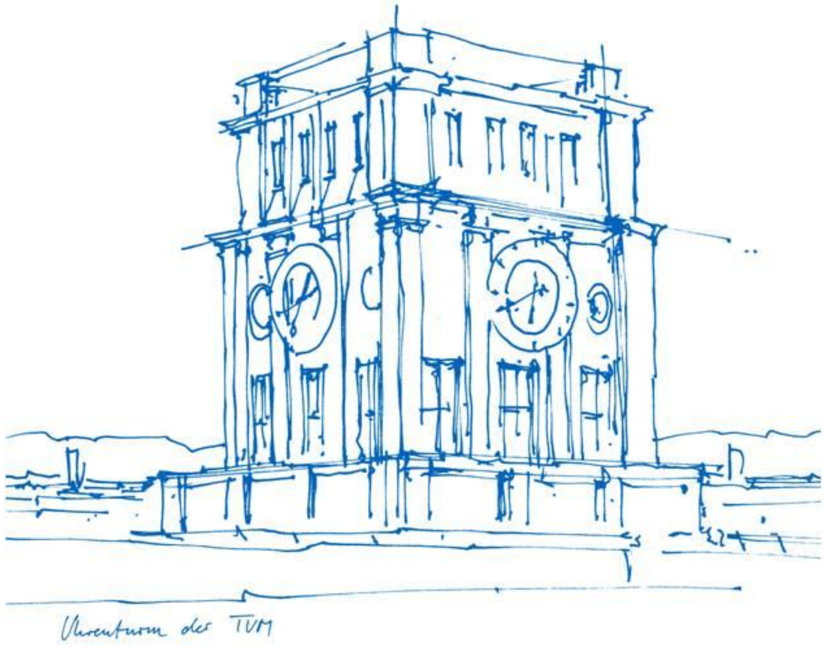
Authors: Zachary Teed, Jia Deng *(Princeton University)*

Witold Pacholarz

The Evolution of Motion Estimation and Real-time 3D Reconstruction

Technical University of Munich

Munich, 25 January 2022



2021

Differentiable
Recurrent
Optimization-
I nspired
Design

2021

Differentiable
Recurrent
Optimization-
I nspired
Design

- 3 sensor modalities
- 4 datasets

2021

Differentiable
Recurrent
Optimization-
I nspired
Design

- 3 sensor modalities
- 4 datasets

SOTA in each case

2021

Differentiable
Recurrent
Optimization-
I nspired
Design

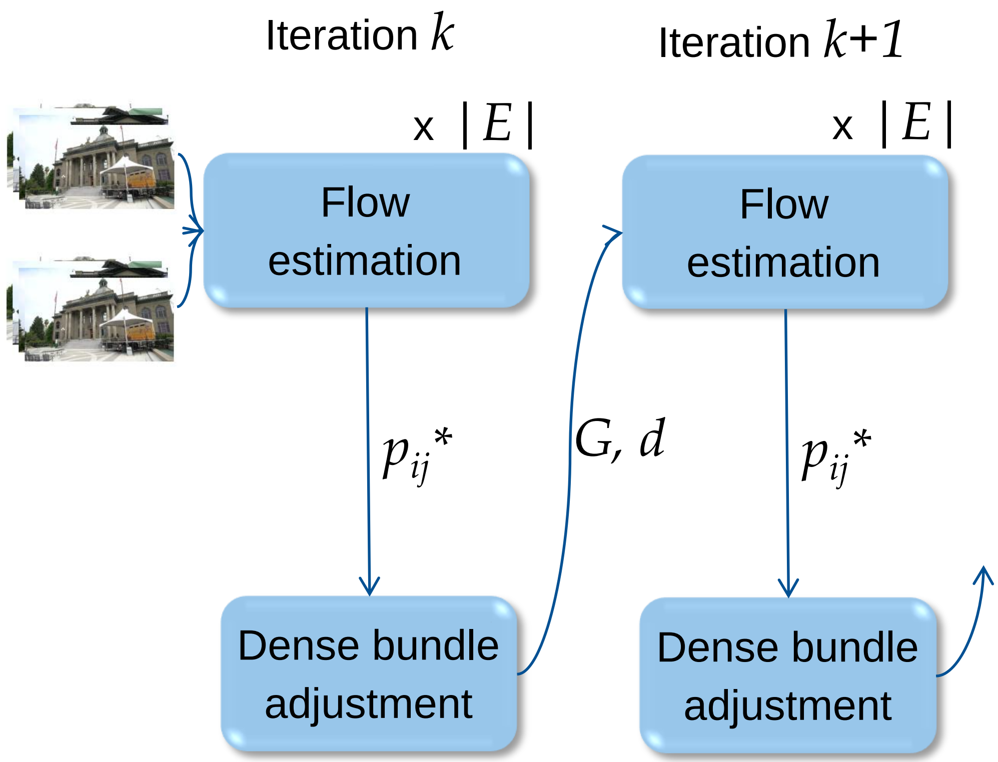
Source: www.theatlantic.com

Agenda

1. Introduction
2. Overview
3. Comparison with similar DL-based methods
4. Method description
5. Experiments and results
6. Personal comments
7. Summary
8. Discussion

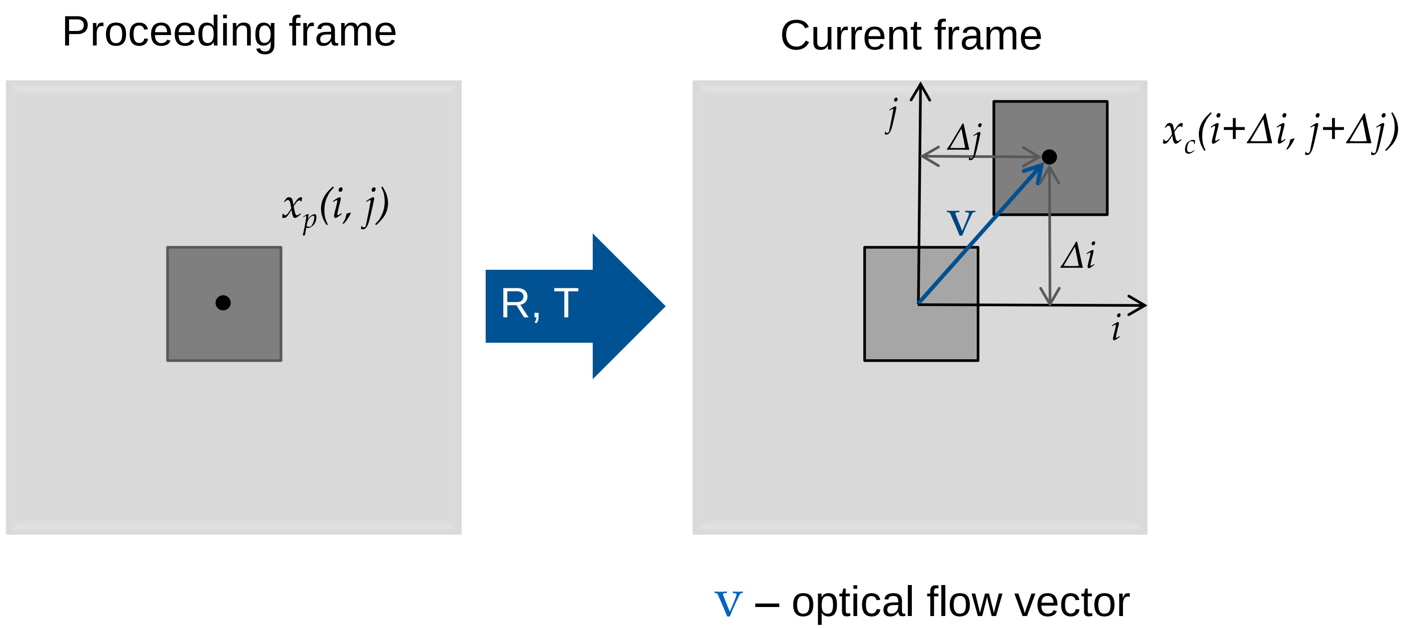
Main idea

- Builds upon the neural network-based model for optical flow estimation called RAFT
  „RAFT: Recurrent All-Pairs Field Transforms for Optical Flow"; Zachary Teed, Jia Deng; 2020
- Leverages a Dense Bundle Adjustment layer to get updated poses and depth
- End-to-end differentiable approach, bundle adjustment used during training



Optical flow estimation

- Optical flow relates to apparent 2D motion observable between consecutive camera frames
- The Lucas & Kanade and Horn & Schunk methods are well-known traditional approaches for flow etimation. However, they are mostly limited to small deformations *(Source: Computer Vision 2; D. Cremers; 2021)*

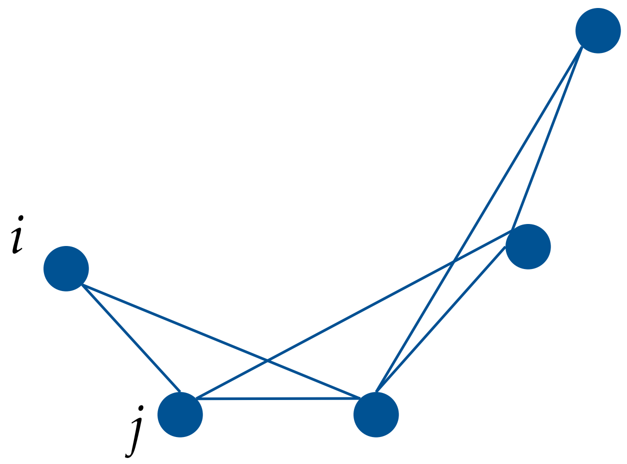


Key aspects

- Optimizes pixel-wise geometric reprojection error
- There is no preprocessing step to detect and match features
- Uses a frame graph to encode the co-visibility between frames
- Performs global bundle adjustment for the entire history of keyframes, assuring loop closure

Frame graph

$$(\mathcal{V}, \mathcal{E}) \quad (i, j) \in \mathcal{E}$$



Comparison with similar DL-based approaches

BA-Net

- Optimizes photometric reprojection error
- Optimizes on few coefficients
- Limited SLAM performance

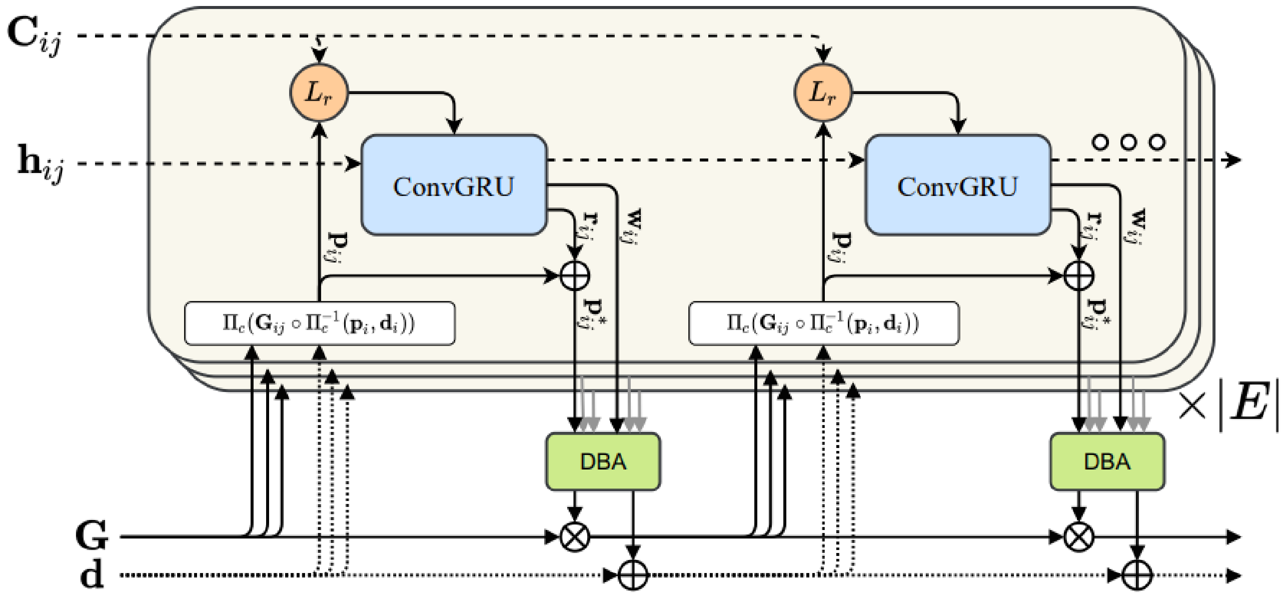
„BA-Net: Dense Bundle Adjustment Network";
Chengzhou Tang, Ping Tan; 2019

DeepFactors

- Jointly optimizes pose and depth
- Optimizes parameters of a learned depth basis
- Capable of loop closure

„DeepFactors: Real-Time Probabilistic Dense Monocular SLAM";
J. Czarnowski at al.; 2020

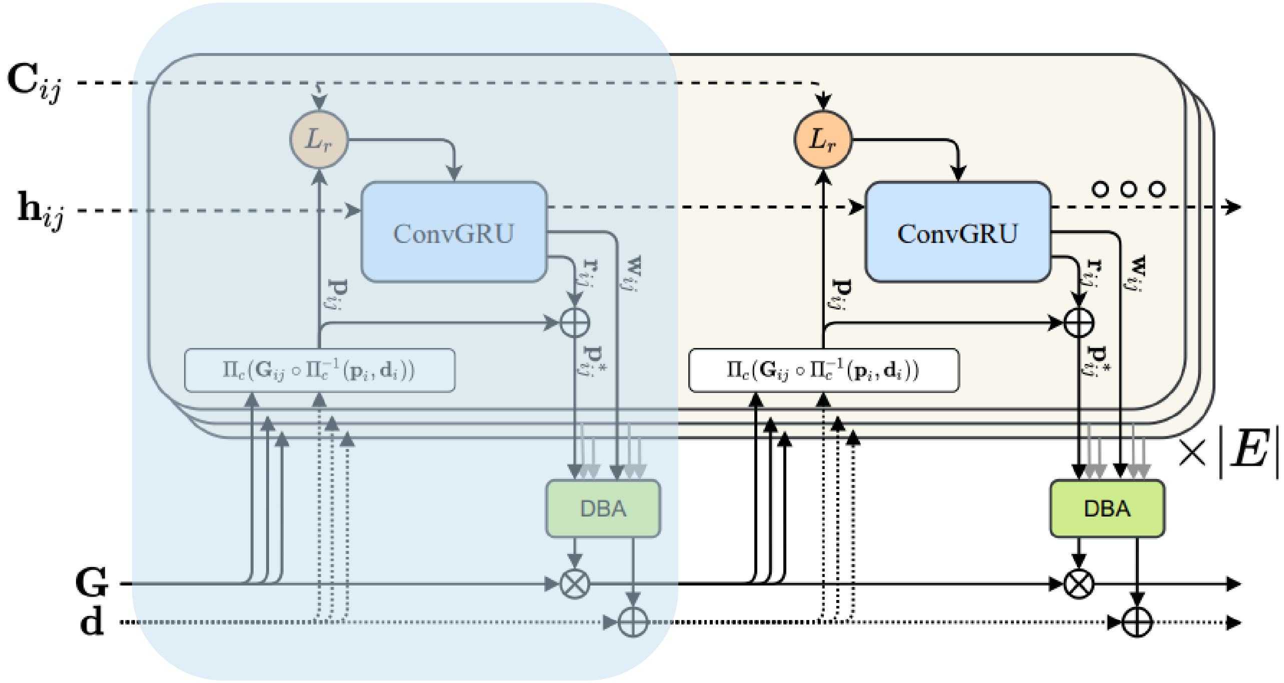
Sequential update operators



Gated
Recurrent
Unit

- Mechanism in Recurrent Neural Networks involving gates
- Good for long-term dependencies as it helps avoid vanishing gradients
- ConvGRU leverages convolutions

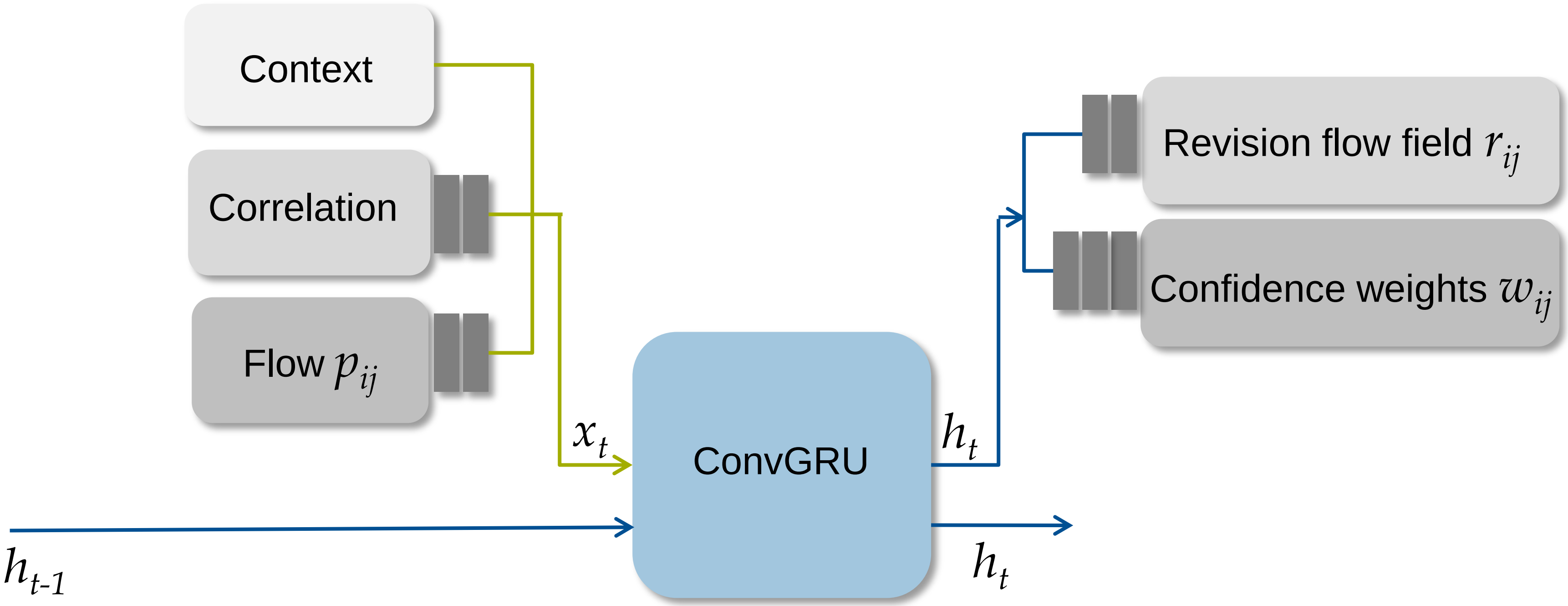
Sequential update operators



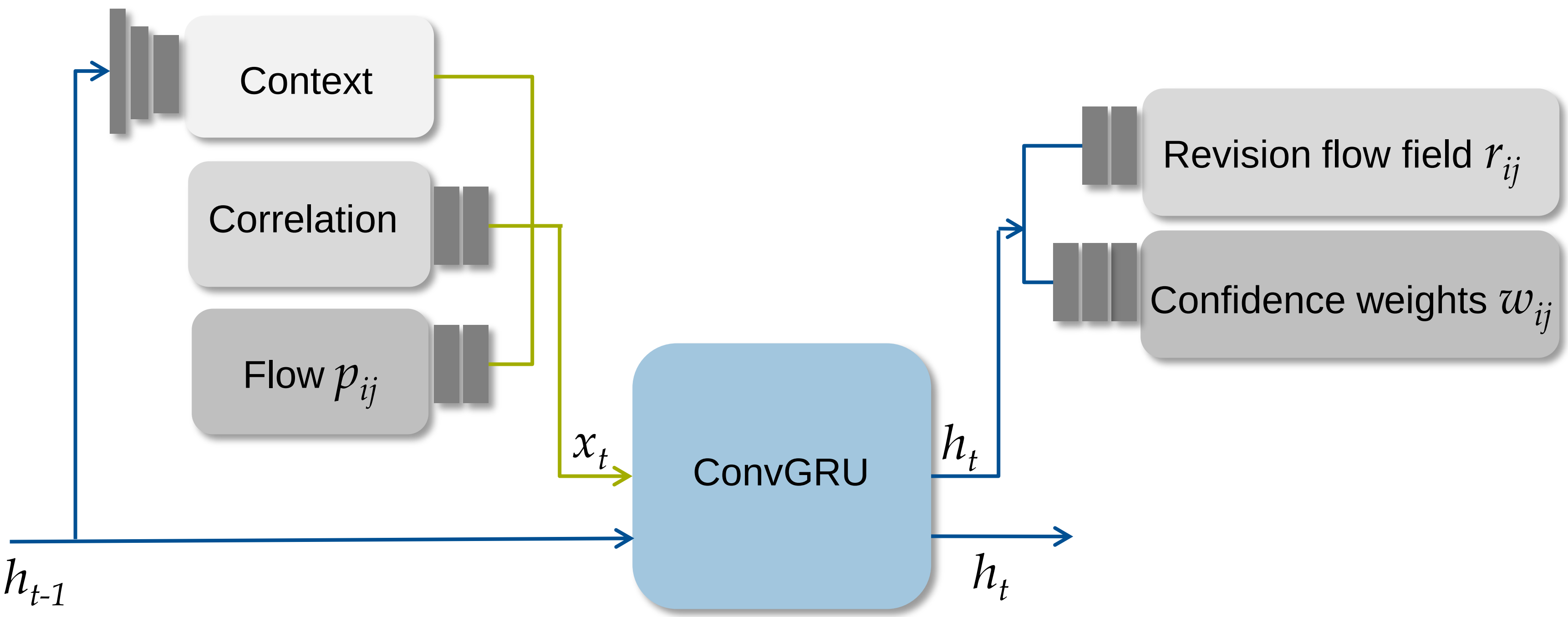
Gated
Recurrent
Unit

- Mechanism in Recurrent Neural Networks involving gates
- Good for long-term dependencies as it helps avoid vanishing gradients
- ConvGRU leverages convolutions

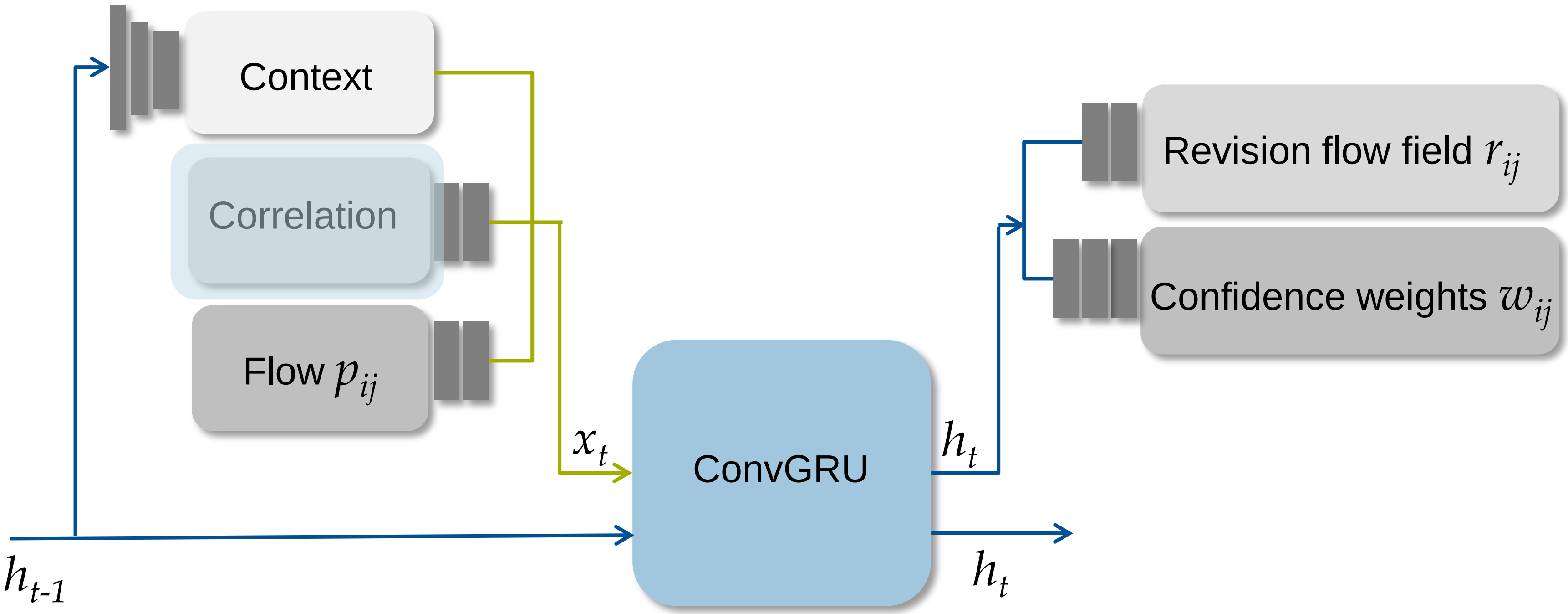
Update operator architecture



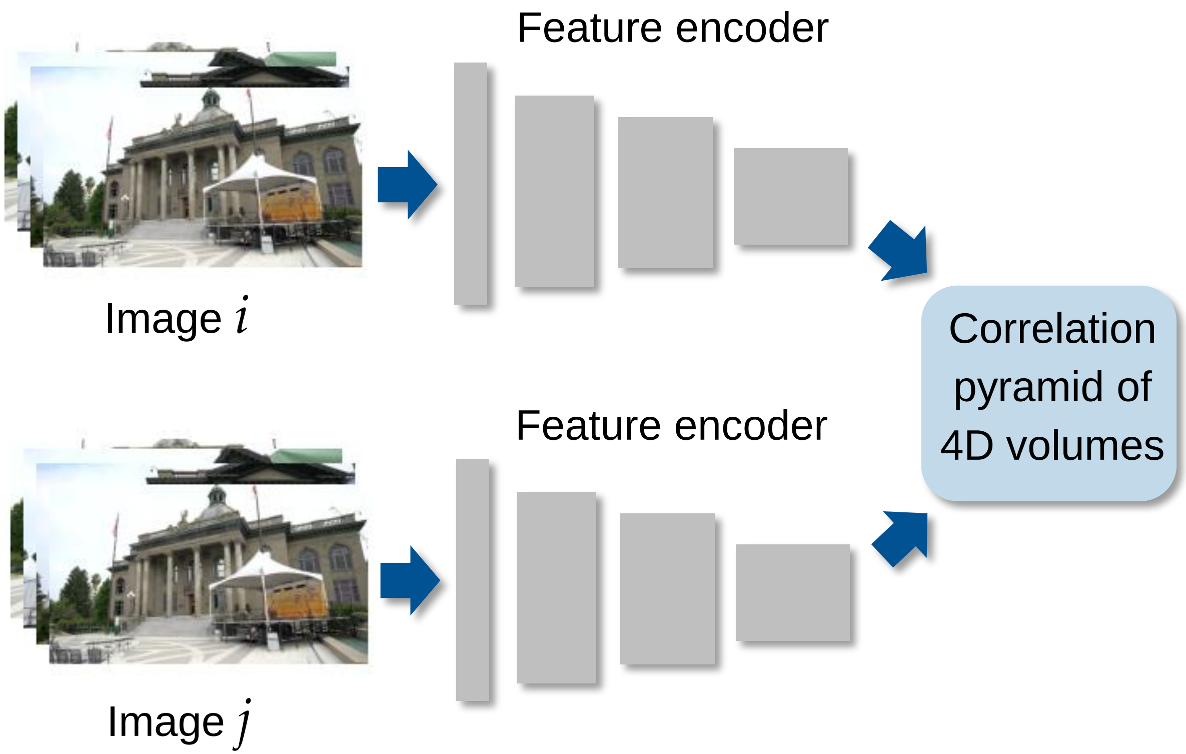
Update operator architecture



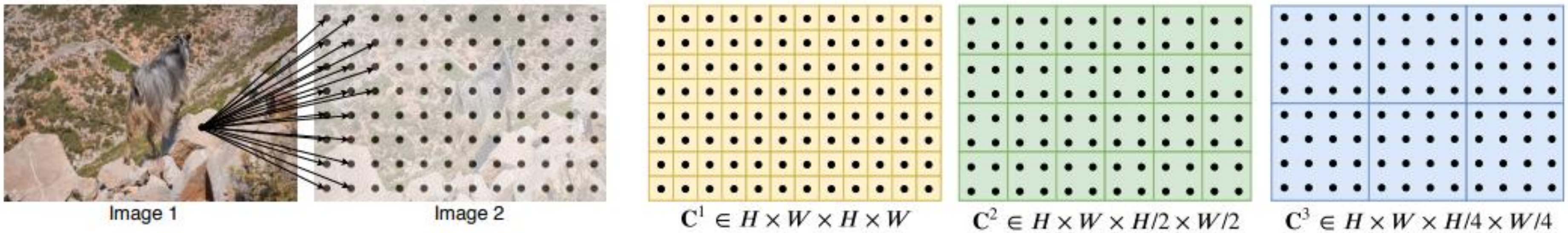
Update operator architecture



Context and feature encoder



Correlation pyramid



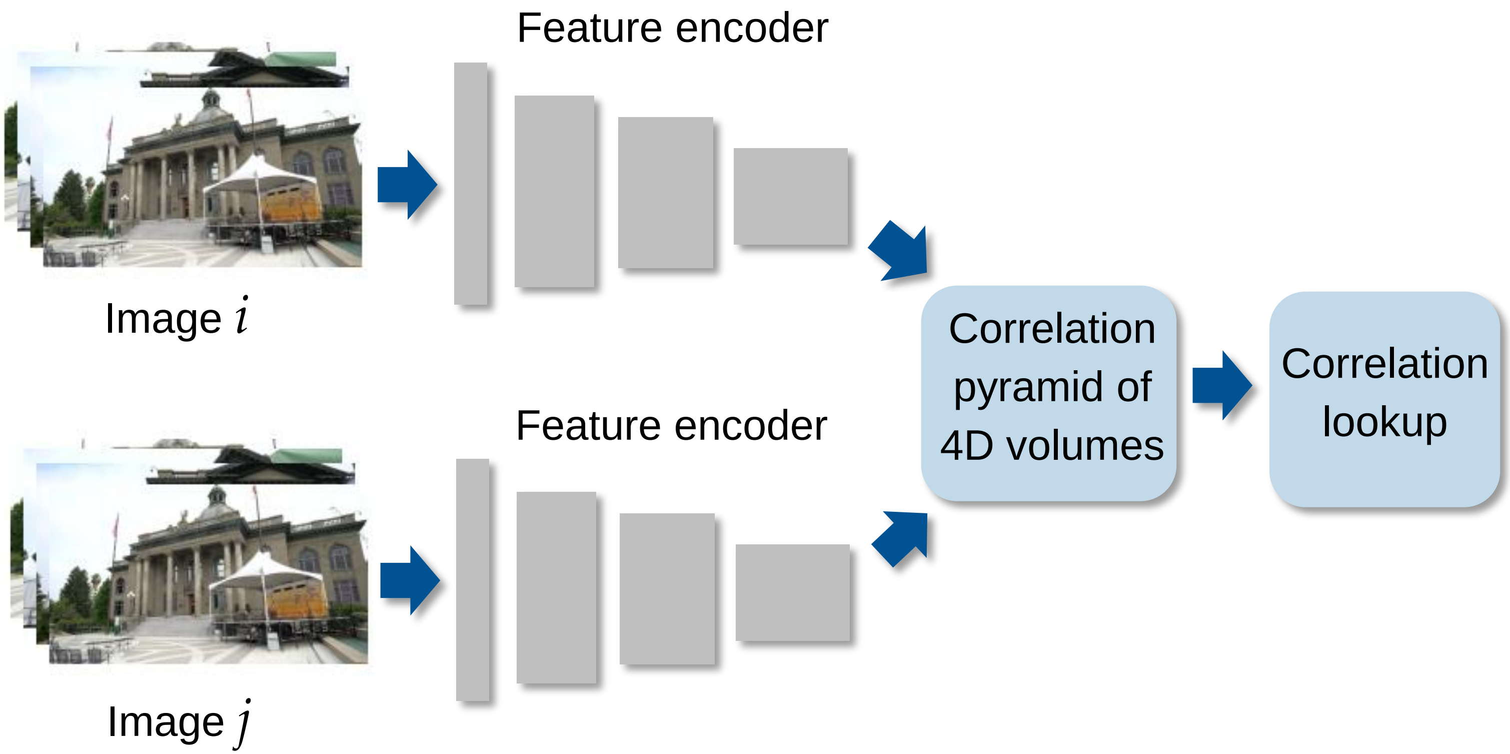
- 4D correlation volume is computed as a dot product of all pairs of vectors of extracted features from two images

$$\mathbf{C}(g_\theta(I_1), g_\theta(I_2)) \in \mathbb{R}^{H \times W \times H \times W}$$

- Average pooling is performed over last 2 dimensions
- Result: 4-level correlation pyramid

$$\mathbf{C}^k \qquad H \times W \times H/2^k \times W/2^k$$

Context and feature encoder



Correlation Lookup

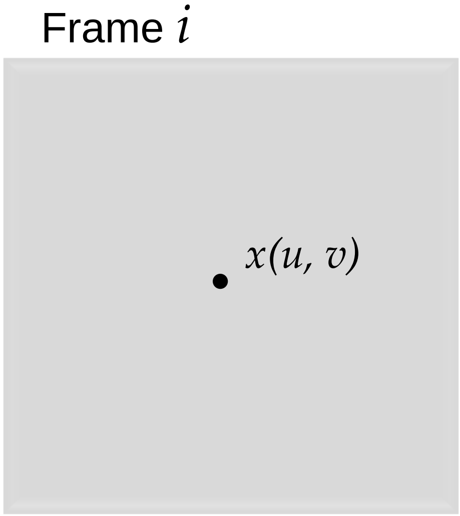
- Use current optical flow p_{ij} and correlation pyramid
- Map each pixel in I_i to its estimated correspondence in I_j
- Local grid around x'

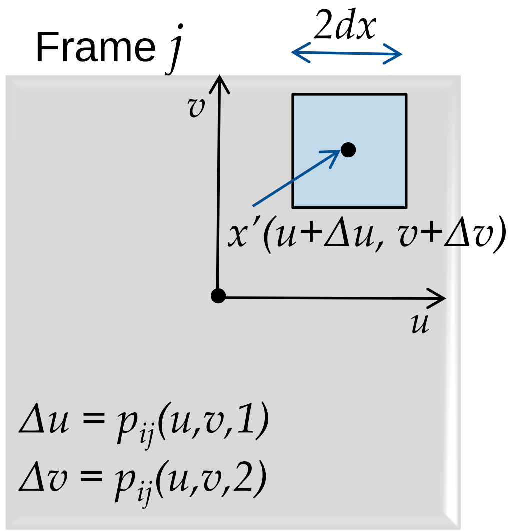
$$\mathcal{N}(\mathbf{x}')_r = \{\mathbf{x}' + \mathbf{dx} \mid \mathbf{dx} \in \mathbb{Z}^2, ||\mathbf{dx}||_1 \leq r\}$$

- Lookups performed on each level of the correlation pyramid

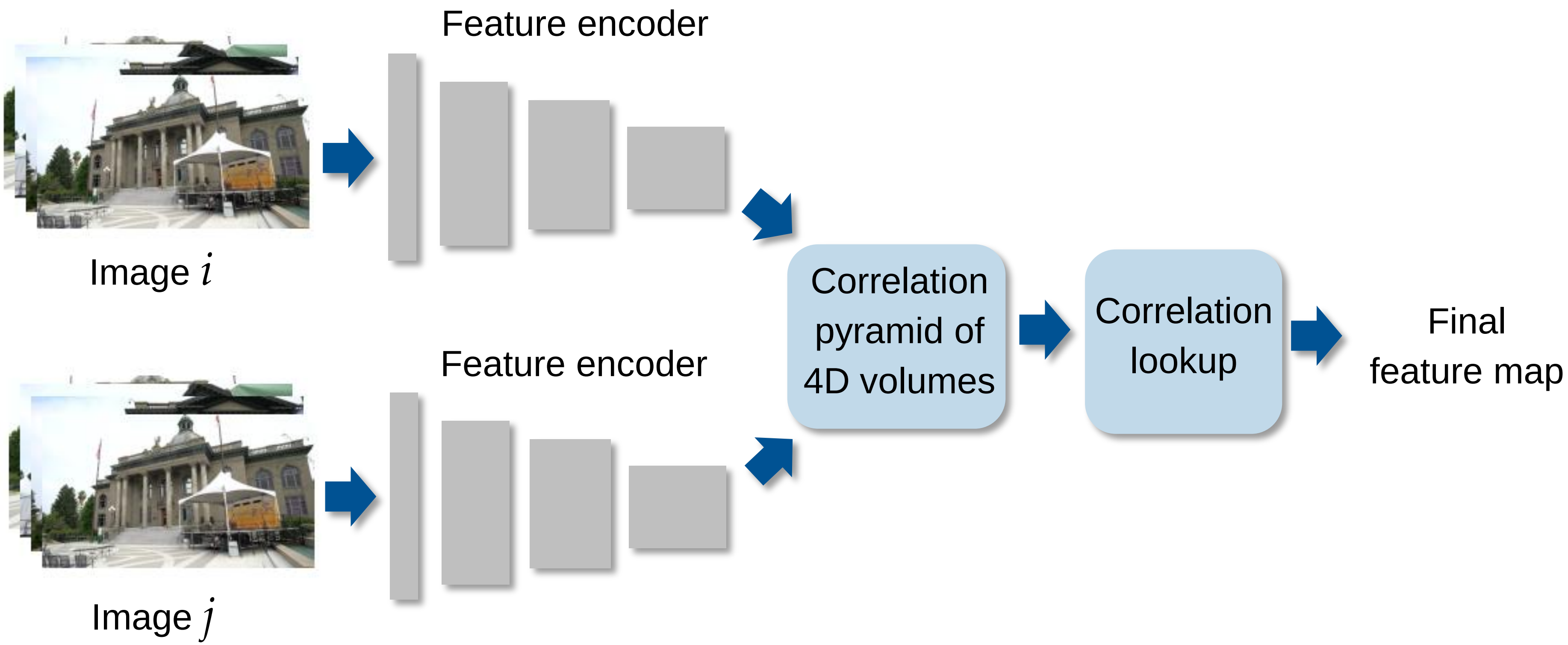
$$\mathcal{N}(\mathbf{x}'/2^k)_r$$

- Larger context at lower levels
- Concatenate values from each level into a single feature map

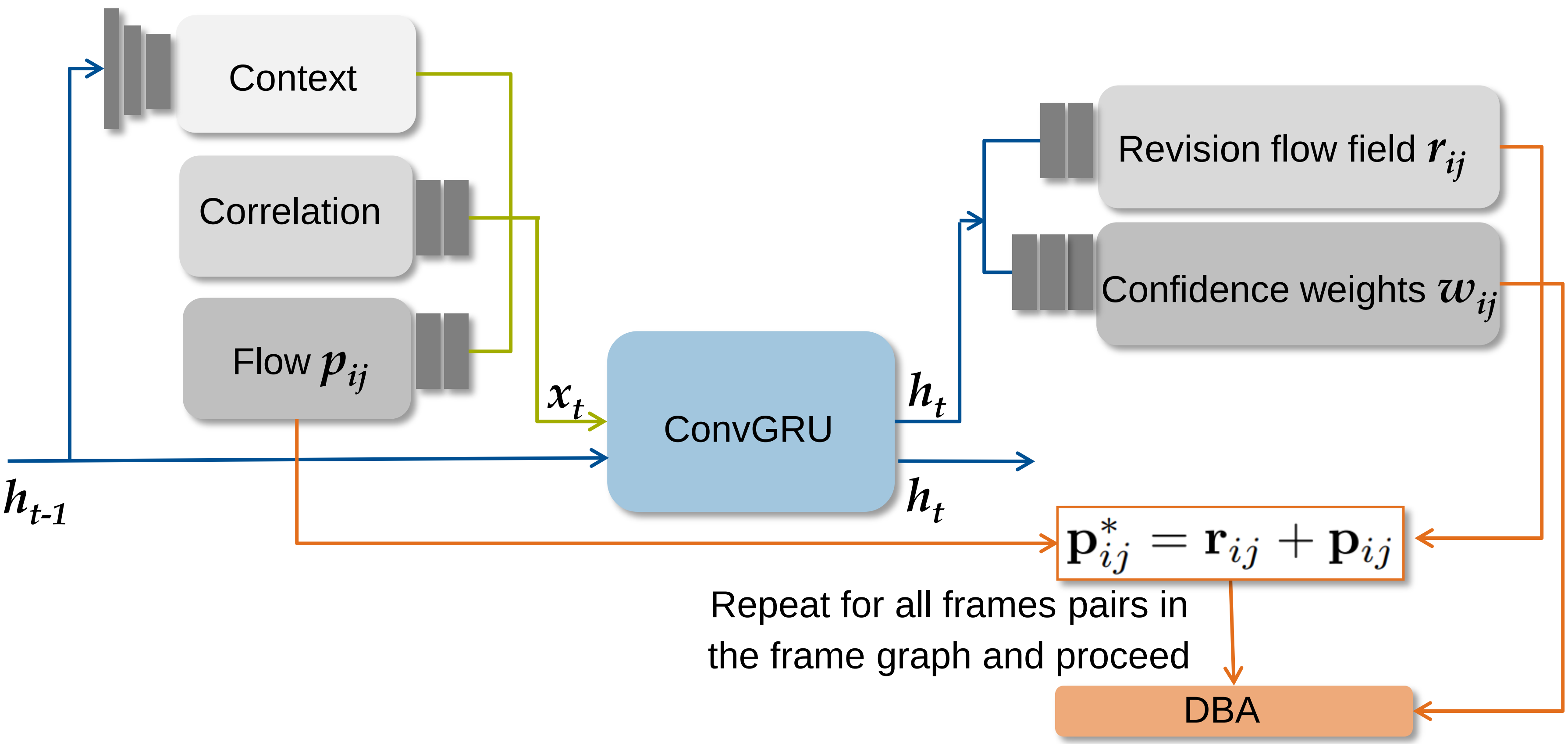




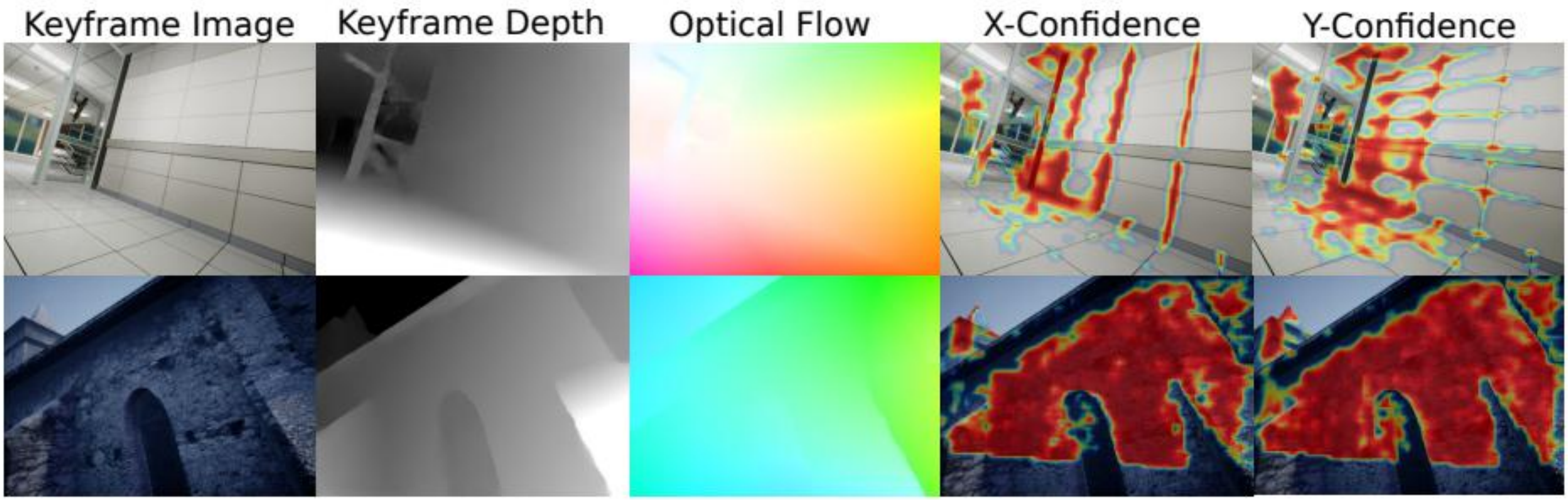
Context and feature encoder



Update operator architecture



Visualizations



Dense Bundle Adjustment layer (DBA)

- Pose and pixelwise depth updates
- Mahalanobis distance weighting error terms

$$\mathbf{E}(\mathbf{G}', \mathbf{d}') = \sum_{(i,j)\in\mathcal{E}} \left\| \mathbf{p}^*_{ij} - \Pi_c(\mathbf{G}'_{ij} \circ \Pi_c^{-1}(\mathbf{p}_i, \mathbf{d}'_i)) \right\|^2_{\Sigma_{ij}}$$

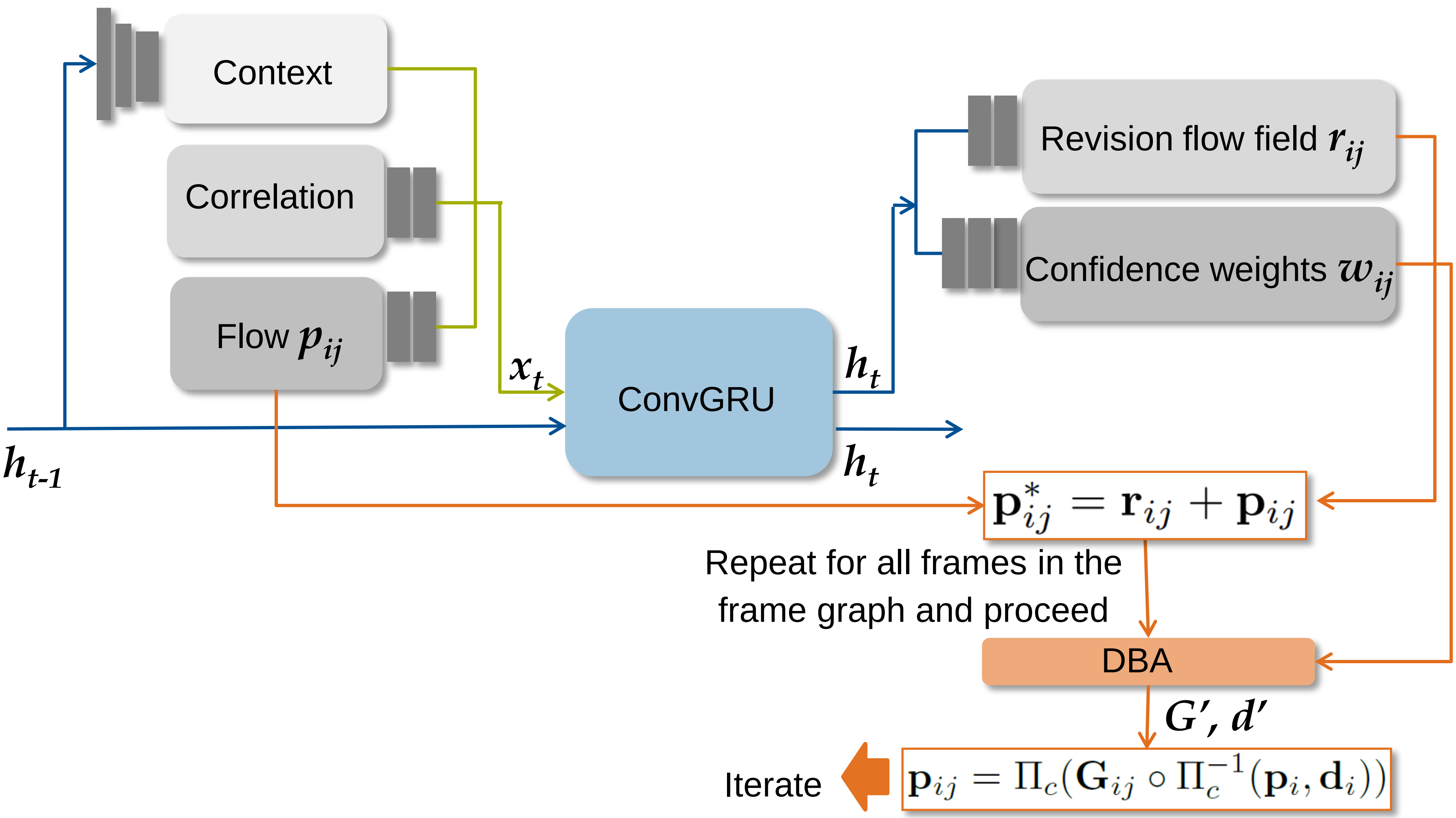
$$\Sigma_{ij} = \operatorname{diag} \mathbf{w}_{ij}$$

- Gauss-Newton algorithm
- Schur complement to get the updates

$$\mathbf{G}^{(k+1)} = \operatorname{Exp}(\Delta\boldsymbol{\xi}^{(k)}) \circ \mathbf{G}^{(k)}, \qquad \mathbf{d}^{(k+1)} = \Delta\mathbf{d}^{(k)} + \mathbf{d}^{(k)}$$

- Backpropagation through the layer during training

Update operator architecture



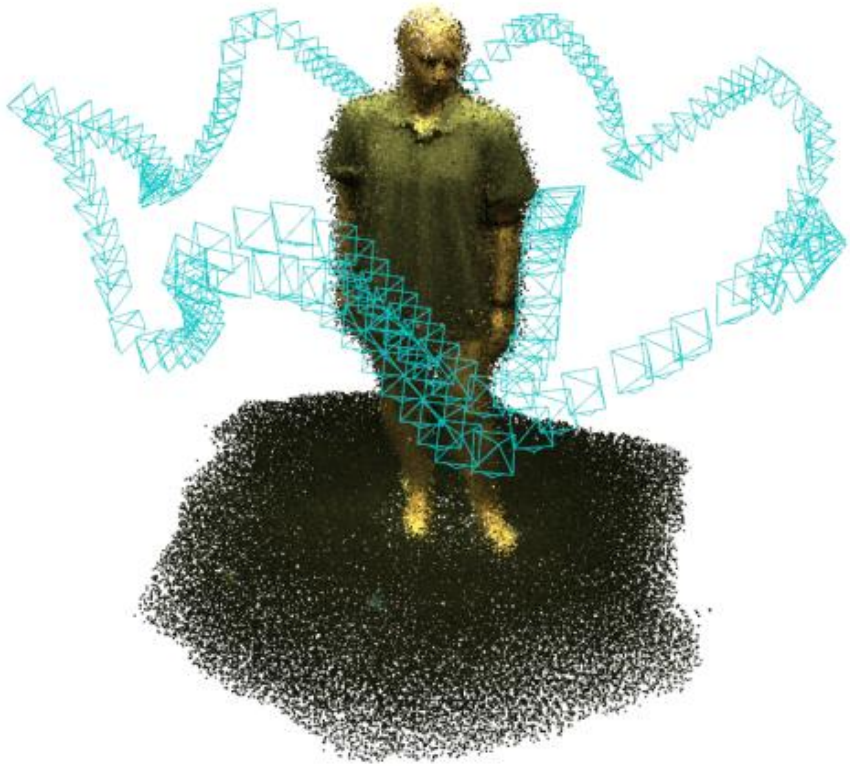
Network supervision

- Network loss is a composition of flow loss and pose loss

- Flow loss is calculated for adjacent frames in the form of the average L2 distance between two correspondence fields

- The pose loss is the distance between the predicted and ground truth poses

$$\mathcal{L}_{pose} = \sum_i || \mathrm{Log}_{SE3}(\mathbf{T}_i^{-1} \cdot \mathbf{G}_i)||_2$$



SLAM System

GPU 1

Initialization:
- Set of 12 frames
- Edges between 5 consecutive keyframes
- Run several iterations of the update operator

Frontend:
- Take in new frames
- Extract features
- Compute flow
- Select keyframes
- Perform local bundle adjustment

GPU 2

Backend:
- Global bundle adjustment over the whole history of keyframes
- Loop closure

Absolute Trajectory Error on TartanAir

- TartanAir is a synthetic dataset
- Robustness (no failures) and significantly lowered accuracy

Monocular	MH000	MH001	MH002	MH003	MH004	MH005	MH006	MH007	Avg
ORB-SLAM [31]	1.30	**0.04**	2.37	2.45	X	X	21.47	2.73	-
DeepV2D [48]	6.15	2.12	4.54	3.89	2.71	11.55	5.53	3.76	5.03
TartanVO [54]	4.88	0.26	2.00	0.94	1.07	3.19	1.00	2.04	1.92
Ours	**0.08**	0.05	**0.04**	**0.02**	**0.01**	**1.31**	**0.30**	**0.07**	**0.24**

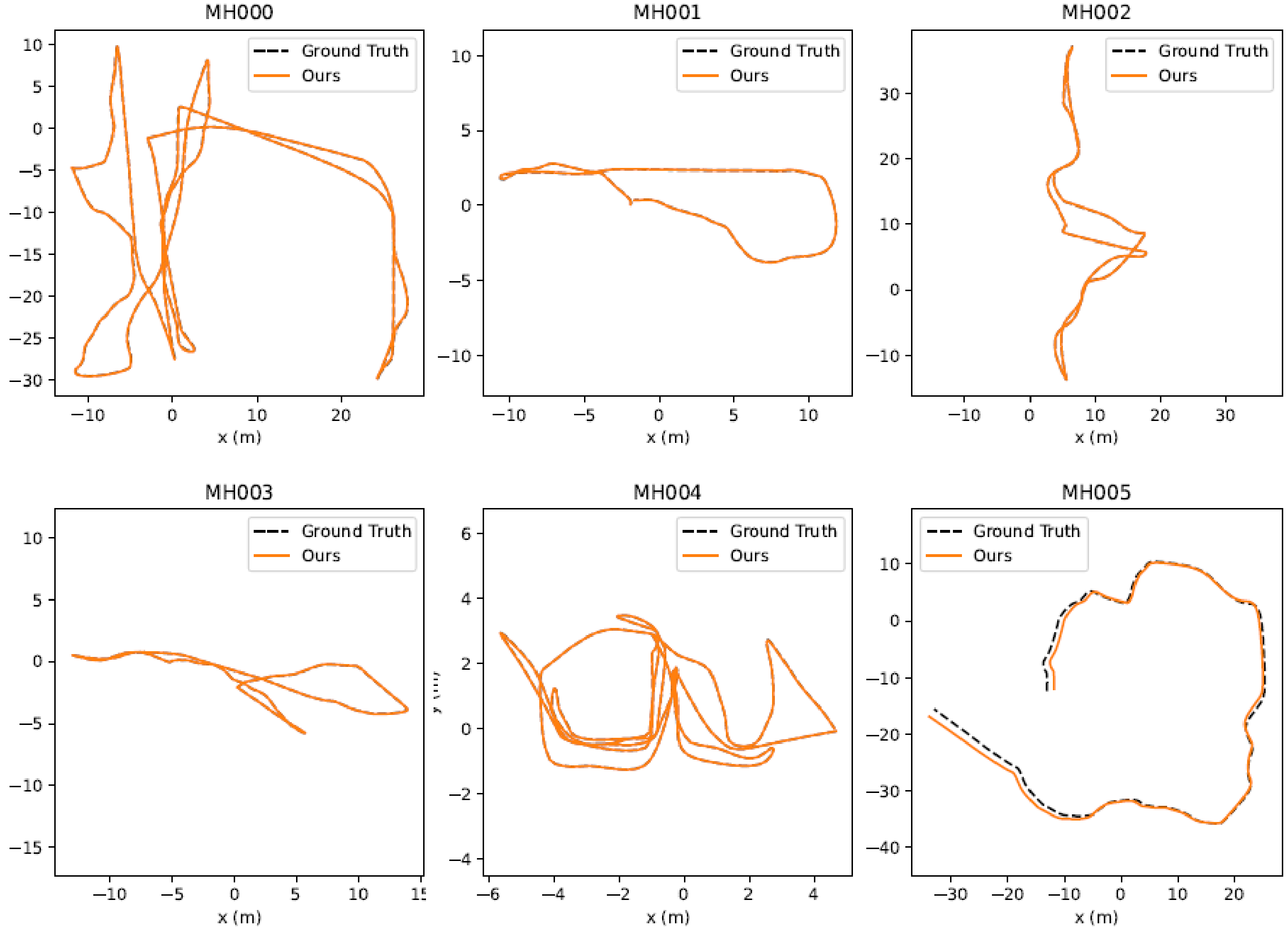
Table 1: Results on the TartanAir monocular benchmark.

Stereo	SH000	SH001	SH002	SH003	SH004	SH005	SH006	SH007	Avg
ORB-SLAM2 [32]	0.05	6.67	X	X	X	X	0.10	X	-
TartanVO [54]	2.52	1.61	3.65	0.29	3.36	4.74	3.72	3.06	2.87
Ours	**0.03**	**0.05**	**0.04**	**0.01**	**0.11**	**0.20**	**0.05**	**0.01**	**0.06**

Table 2: Results on the TartanAir stereo benchmark.

REMARK: for all the evaluations presented on this and the following slides the network was trained only on monocular TartanAir

TartanAir Camera Trajectories



Evauation on TUM-RGBD and ETH-3D SLAM

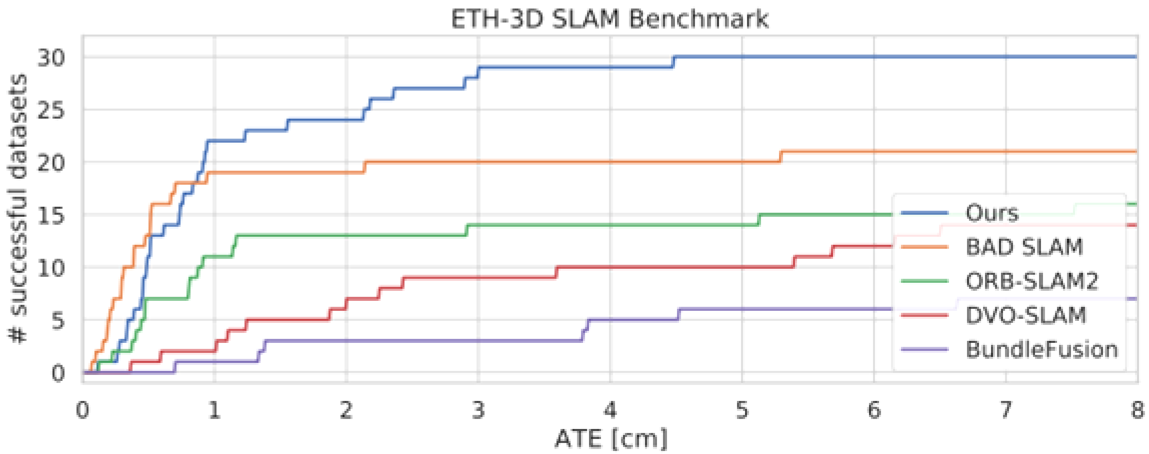
TUM-RGBD

- Challenging dataset for monocular approaches because of heavy rotation, motion blur, rolling shutter

	360	desk	desk2	floor	plant	room	rpy	teddy	xyz	avg
ORB-SLAM2 [32]	X	0.071	X	0.023	X	X	X	X	0.010	-
ORB-SLAM3 [5]	X	**0.017**	0.210	X	0.034	X	X	X	**0.009**	-
DeepTAM[1] [60]	0.111	0.053	0.103	0.206	0.064	0.239	0.093	0.144	0.036	0.116
TartanVO[2] [54]	0.178	0.125	0.122	0.349	0.297	0.333	0.049	0.339	0.062	0.206
DeepV2D [48]	0.243	0.166	0.379	1.653	0.203	0.246	0.105	0.316	0.064	0.375
DeepV2D (TartanAir)	0.182	0.652	0.633	0.579	0.582	0.776	0.053	0.602	0.150	0.468
DeepFactors [9]	0.159	0.170	0.253	0.169	0.305	0.364	0.043	0.601	0.035	0.233
Ours	**0.111**	0.018	**0.042**	**0.021**	**0.016**	**0.049**	**0.026**	**0.048**	0.012	**0.038**

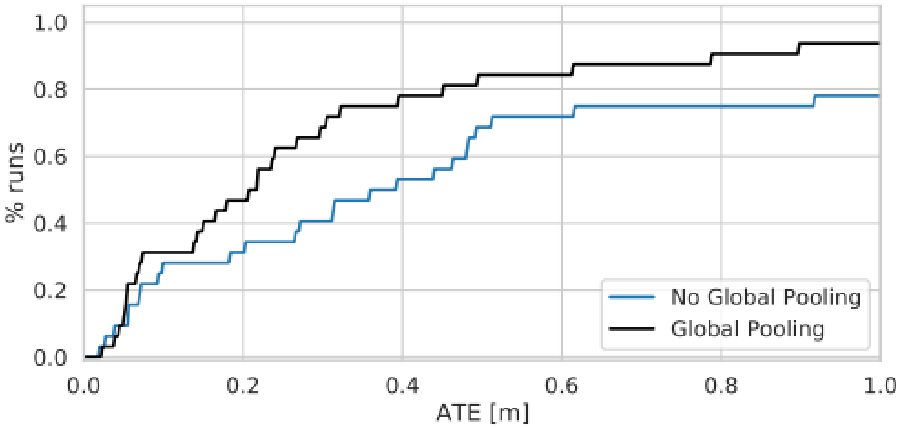
ETH3D-SLAM

- Succesfully tracks 30/32 sequences.



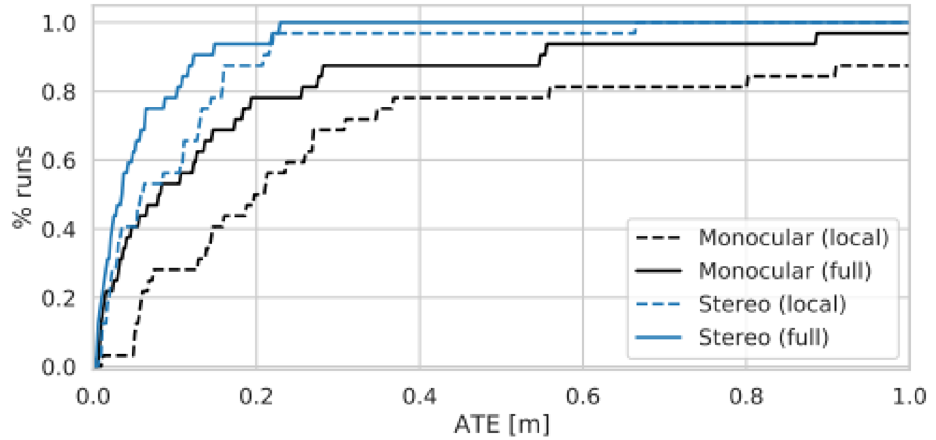
Ablation study

Impact of global context



• The study confirms that global context is a valuable factor for the system performance

Influence of input data and global bundle adjustment



• It can be observed that the model profits both from stereo data and global bundle adjustment

Personal comments / possible improvements

Issue 1

• Due to large resource requirements, the model is trained on low-resolution video which may result in low-quality reconstruction

• Because of the system being computation-heavy, it is not able to run in real-time on TartanAir

Possible solution: *test sparser frame associations in the frame graph to reduce computations and allow higher-resolution data*

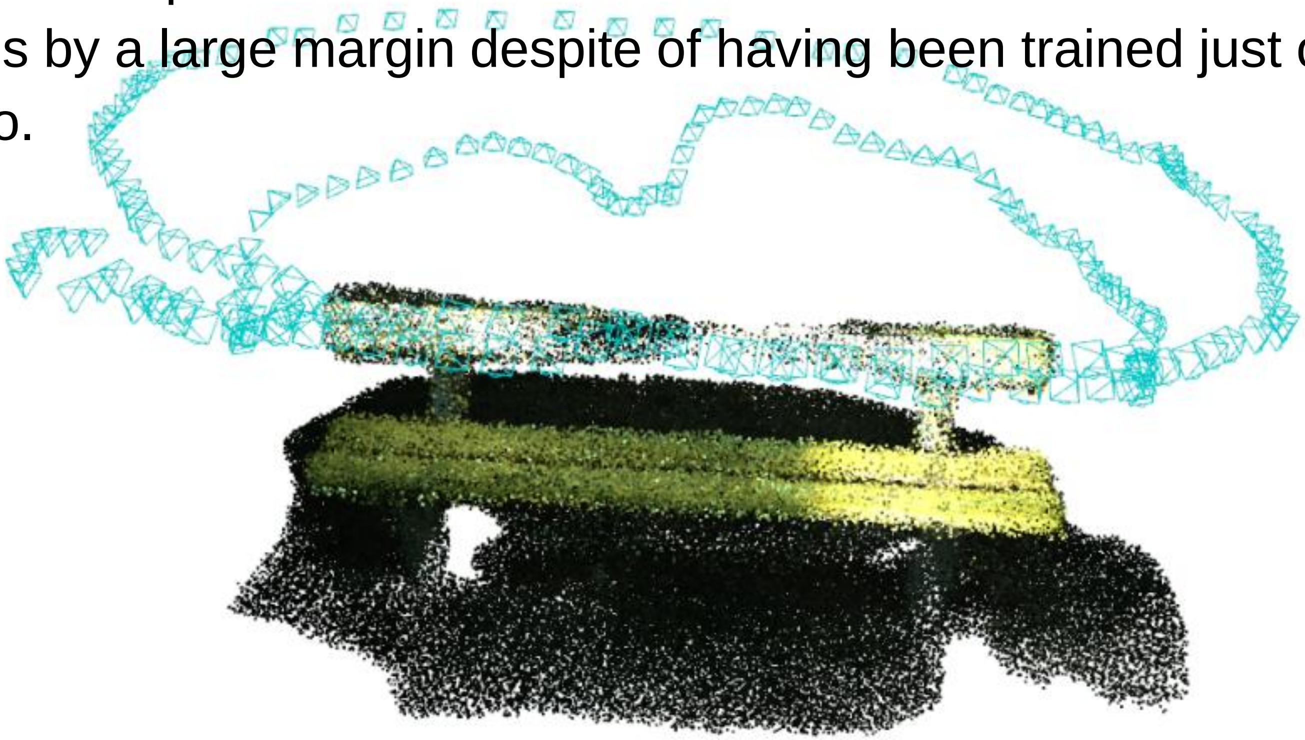
Issue 2

• Accuracy could be improved for the cases in which loop closure is not performed (visible drift on TartanAir trajectories)

Possible solution: *it was shown that stereo video w/o BA led to higher accuracy than monocular video with BA – this could serve as the starting point (e.g. virtual stereo term as in DVSO)*

Personal comments continued

I am particularly impressed by the generalization capabilities of the DROID-SLAM as it outperforms well-established SLAM models on all the tested modalities by a large margin despite of having been trained just on monocular video.



Summary

- DROID-SLAM is currently the state-of-the-art deep learning-based Visual SLAM approach for monocular, stereo and RGB-D data

- Uses end-to end differentiable architecture

- Iteratively estimates optical flow and computes dense bundle adjustment to update poses and depth

- Performs global bundle adjustment to refine results and assure loop closure

Main advantages

High accuracy

- Significantly reduced error on top benchmarks

High robustness

- More succesfully tracked datasets

Strong generalization

- After training on a monocular dataset, it generalizes to stereo and RGB-D data

?

Bibliography

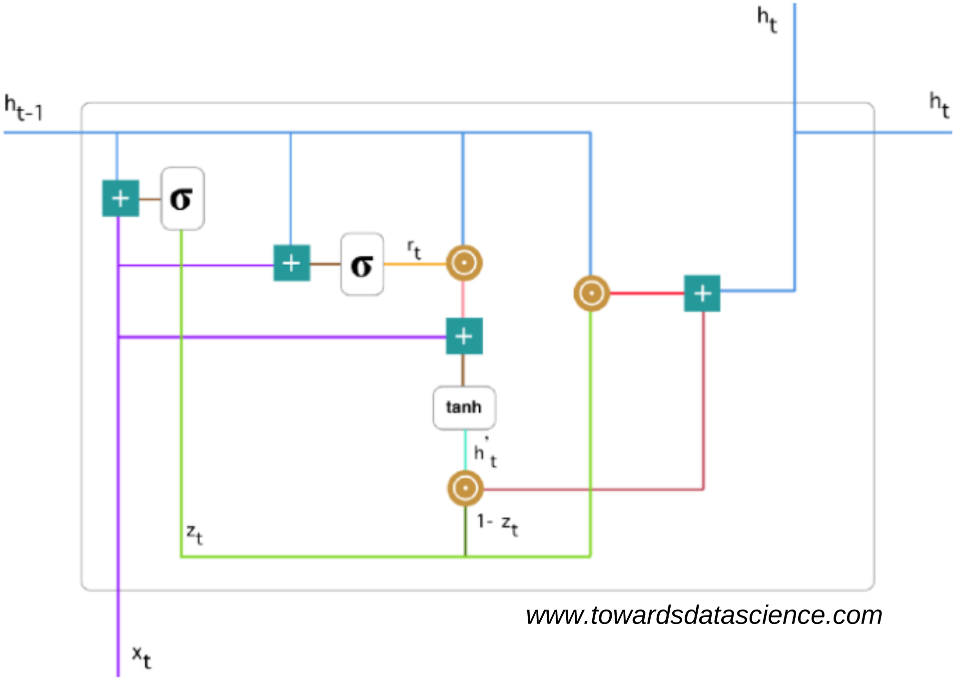
- *„DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras"; Z. Teed, J. Deng; 2021*
- *„RAFT: Recurrent All-Pairs Field Transforms for Optical Flow"; Z. Teed, J. Deng; 2020*
- *„BA-Net: Dense Bundle Adjustment Network"; Chengzhou Tang, Ping Tan; 2019*
- *„DeepFactors: Real-Time Probabilistic Dense Monocular SLAM"; J. Czarnowski at al.; 2020*
- *„Deep Virtual Stereo Odometry: Leveraging Deep Depth Prediction for Monocular Direct Sparse Odometry"; N. Yang at al.; 2018*
- *Computer Vision 2 slides; D. Cremers; 2021*
- *https://github.com/princeton-vl/DROID-SLAM (demo)*
- *www.towardsdatascience.com (GRU architecture)*
- *www.theatlantic.com (DROID photo)*

Extension (for potential questions)

Gated Recurrent Unit

Gated
Recurrent
Unit

- mechanism in Recurrent Neural Networks involving gates
- update gate and reset gate
- good for long-term dependencies
- helps avoid vanishing gradients



www.towardsdatascience.com

$$z_t = \sigma(W^{(z)} x_t + U^{(z)} h_{t-1})$$

$$r_t = \sigma(W^{(r)} x_t + U^{(r)} h_{t-1})$$

$$h_t' = \tanh(W x_t + r_t \odot U h_{t-1})$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot h_t'$$

Convolutional GRU

Gated
Recurrent
Unit

- mechanism in Recurrent Neural Networks involving gates
- update gate and reset gate
- good for long-term dependencies
- helps avoid vanishing gradients

$$z_t = \sigma(\mathrm{Conv}_{3\mathrm{x}3}([h_{t-1}, x_t], W_z))$$

$$r_t = \sigma(\mathrm{Conv}_{3\mathrm{x}3}([h_{t-1}, x_t], W_r))$$

$$\tilde{h_t} = \tanh(\mathrm{Conv}_{3\mathrm{x}3}([r_t \odot h_{t-1}, x_t], W_h))$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h_t}$$

Feature and context encoder