Taking a Deeper Look at the Inverse Compositional Algorithm

Zhaoyang Lv[1] , Frank Dellaert[1], James M. Rehg[1], Andreas Geiger[2]

[1]Georgia Institute of Technology, Institute for Robotics and Intelligent Machines
[2]Autonomous Vision Group, MPI-IS and University of Tübingen

Georgia Tech Institute for Robotics and Intelligent Machines

Max Planck Institute for Intelligent Systems Autonomous Vision Group

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

CVPR LONG BEACH CALIFORNIA June 16-20, 2019

Motivation

Problem: Solve the Lucas-Kanade objective $\min_{\boldsymbol{\xi}} \|\mathbf{I}(\boldsymbol{\xi}) - \mathbf{T}(\mathbf{0})\|_2^2$

Classical Solution: The Inverse Compositional (IC) Algorithm [Baker and Matthews, 04]

$$\mathbf{r}_k(\mathbf{0}) = \mathbf{I}(\boldsymbol{\xi}_k) - \mathbf{T}(\mathbf{0})$$

$$\Delta\boldsymbol{\xi} = (\mathbf{J}^T\mathbf{W}\mathbf{J} + \lambda\,\mathrm{diag}(\mathbf{J}^T\mathbf{W}\mathbf{J}))^{-1}\mathbf{J}^T\mathbf{W}\,\mathbf{r}_k(\mathbf{0})$$

$$\boldsymbol{\xi}_{k+1} = \boldsymbol{\xi}_k \circ (\Delta\boldsymbol{\xi})^{-1}$$

Limitations:

- Photometric consistency may break
- Weight matrix only depends on point residual
- Heuristic rule for damping is not optimal

Our Goal: Overcome its limitations **from a learning perspective**.

Related Work: Using data to learn a direct mapping from images to warping parameters using linear approximation [1] or deep neural networks [2].

Contributions

Our Solution: We take a **deeper** look at IC algorithm with three contributions.

(A) Two-view Feature Encoder

(B) Convolutional M-estimator

(C) Trust Region Network

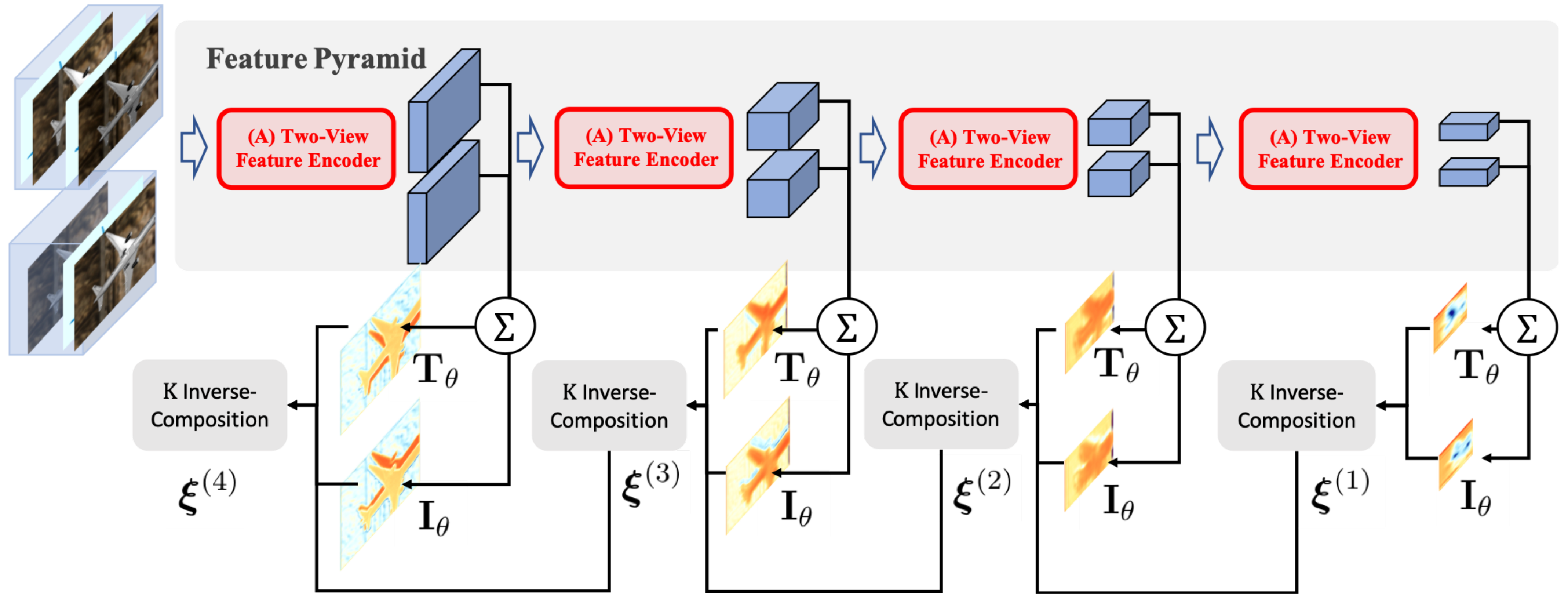
$$\mathbf{r}_k = \mathbf{I}_\theta(\boldsymbol{\xi}_k) - \mathbf{T}_\theta(\mathbf{0})$$

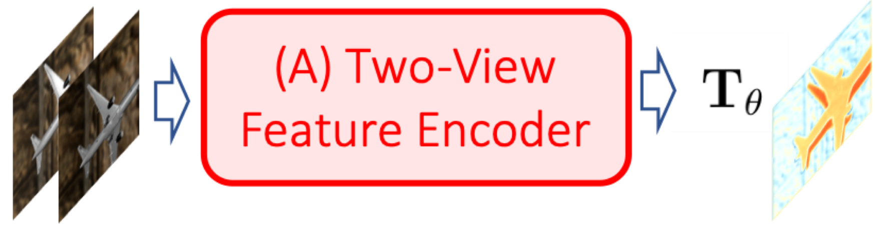
$$\Delta\boldsymbol{\xi} = (\mathbf{J}^T\mathbf{W}_\theta\mathbf{J} + \mathrm{diag}(\boldsymbol{\lambda}_\theta))^{-1}\mathbf{J}^T\mathbf{W}_\theta\mathbf{r}_k(\mathbf{0})$$

$$\boldsymbol{\xi}_{k+1} = \boldsymbol{\xi}_k \circ (\Delta\boldsymbol{\xi})^{-1}$$

Our Method

A Coarse-to-fine Deep Inverse Compositional Algorithm

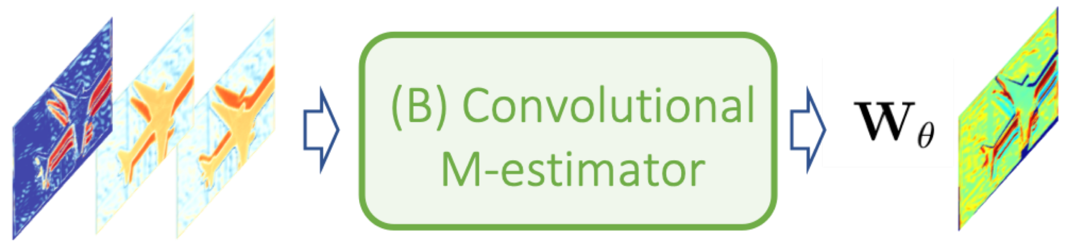




Extract features of template and image:

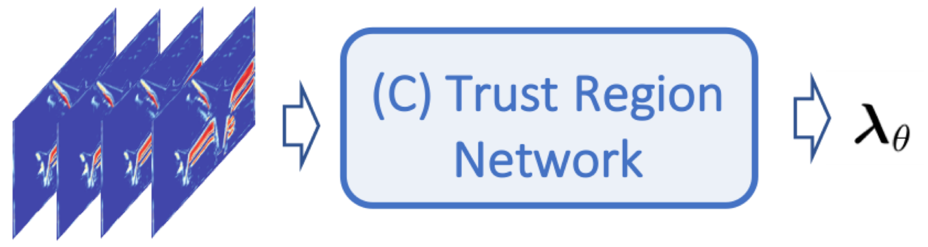
$$\mathbf{I}_\theta = \phi_\theta([\mathbf{I}, \mathbf{T}])$$

$$\mathbf{T}_\theta = \phi_\theta([\mathbf{T}, \mathbf{I}])$$



Estimate weights from features and residuals:

$$\mathbf{W}_\theta = \mathrm{diag}(\psi_\theta(\mathbf{I}_\theta(\boldsymbol{\xi}_k), \mathbf{T}_\theta(\mathbf{0}), \mathbf{r}_k))$$



Calculate residuals from damping proposals:

$$\Delta\boldsymbol{\xi}_i = (\mathbf{J}^T\mathbf{W}\mathbf{J} + \lambda_i\,\mathrm{diag}(\mathbf{J}^T\mathbf{W}\mathbf{J}))^{-1}\mathbf{J}^T\mathbf{W}\,\mathbf{r}_k(\mathbf{0})$$

$$\mathbf{r}_{k+1}^{(i)} = \mathbf{I}_\theta(\boldsymbol{\xi}_k \circ (\Delta\boldsymbol{\xi}_i)^{-1}) - \mathbf{T}_\theta(\mathbf{0})$$

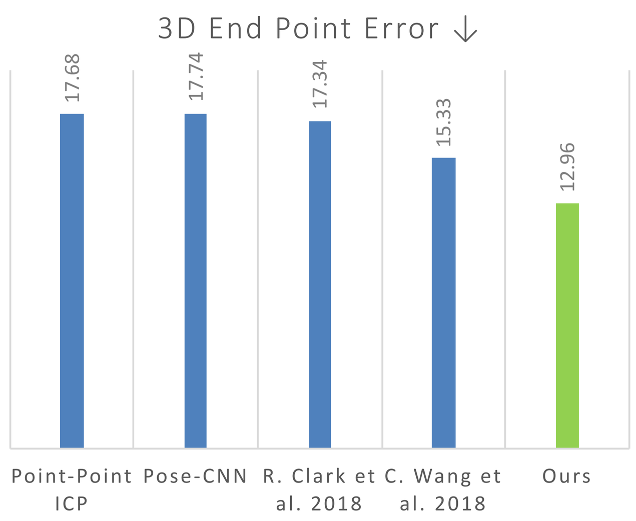
Estimate optimal damping from residual maps:

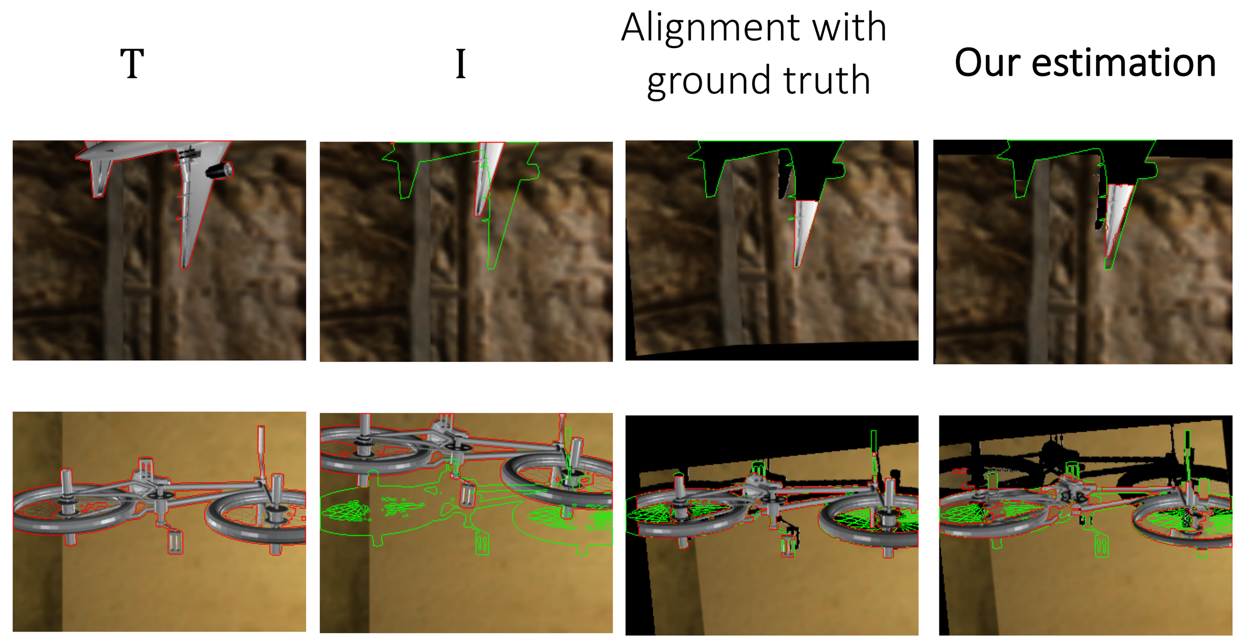
$$\boldsymbol{\lambda}_\theta = \nu_\theta\left(\mathbf{J}^T\mathbf{W}_\theta\mathbf{J}, \left[\mathbf{J}^T\mathbf{W}_\theta\mathbf{r}_{k+1}^{(1)}, \ldots, \mathbf{J}^T\mathbf{W}_\theta\mathbf{r}_{k+1}^{(N)}\right]\right)$$

Experiments

3D Object Motion

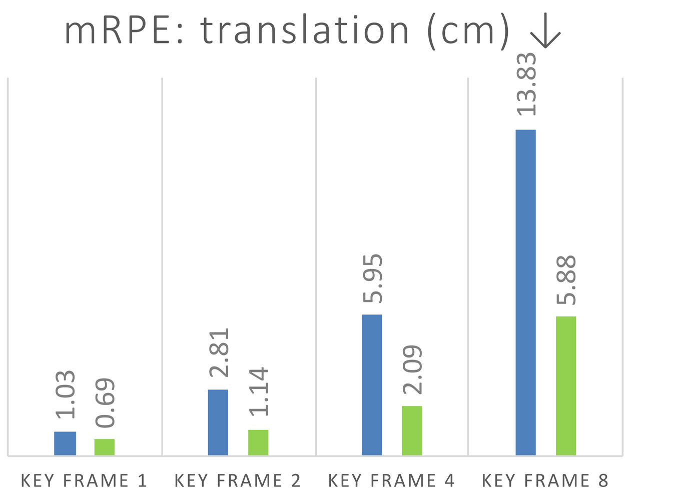
On unseen test objects

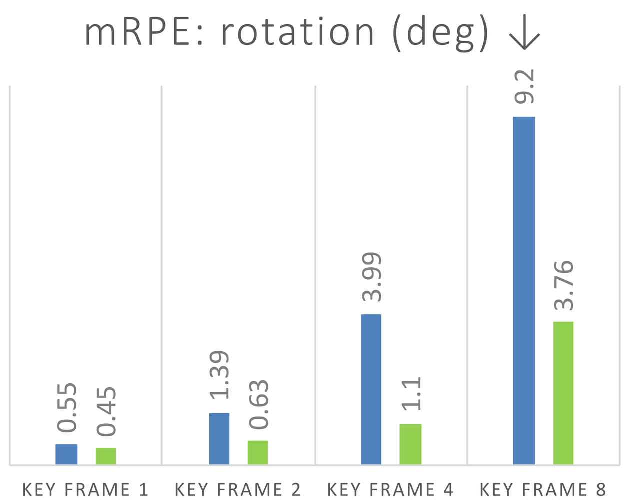




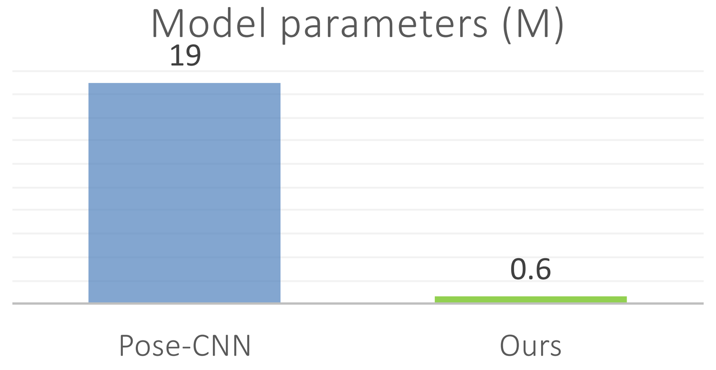
Visual Odometry

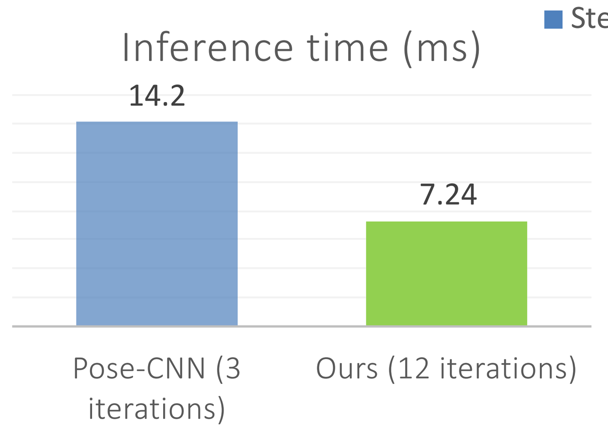
On unseen test trajectories





■ Steinbrücker et al, 2011 ■ Ours





Pose-CNN: a variant of IC-STN [2]

Code

Conclusion

- Our proposed method is **learnable, accurate, small, and fast** in inference.
- It can be a core building block of many applications including object tracking, visual odometry, and 3D reconstruction.

[1] Jurie and Dhome. Hyperplane approximation for template matching. PAMI, 2002
[2] Lin and Lucey. Inverse compositional spatial transformer networks. CVPR, 2017