Taking a Deeper Look at the Inverse Compositional Algorithm

Andreas Geiger

Autonomous Vision Group
University of Tübingen / MPI for Intelligent Systems

June 17, 2018

University of Tübingen
MPI for Intelligent Systems
Autonomous Vision Group

Making **Robust** Image Alignment even more **Robust**

Andreas Geiger

Autonomous Vision Group
University of Tübingen / MPI for Intelligent Systems

June 17, 2018

University of Tübingen
MPI for Intelligent Systems
Autonomous Vision Group

Making **Robust** Image Alignment even more **Robust** but certainly not more **Uncertain**

Andreas Geiger

Autonomous Vision Group
University of Tübingen / MPI for Intelligent Systems

June 17, 2018

University of Tübingen
MPI for Intelligent Systems
Autonomous Vision Group

Making **Robust** Image Alignment even more **Robust** but **certainly** not more **Uncertain**

Andreas Geiger

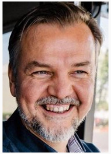
Autonomous Vision Group
University of Tübingen / MPI for Intelligent Systems

June 17, 2018

University of Tübingen
MPI for Intelligent Systems
Autonomous Vision Group

Taking a Deeper Look at the Inverse Compositional Algorithm

[Lv, Dellaert, Rehg & Geiger, CVPR 2019]

A Seminal Paper

A Seminal Paper

An Iterative Image Registration Technique with an Application to Stereo Vision

Bruce D. Lucas
Takeo Kanade

Computer Science Department
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

Abstract

Image registration finds a variety of applications in computer vision. Unfortunately, traditional image registration techniques tend to be costly. We present a new image registration technique that makes use of the spatial intensity gradient of the images to find a good match using a type of Newton-Raphson iteration. Our technique is faster because it examines far fewer potential matches between the images than existing techniques. Furthermore, this registration technique can be generalized to handle rotation, scaling and shearing. We show show our technique can be adapted for use in a stereo vision system.

1. Introduction

Image registration finds a variety of applications in computer vision, such as image matching for stereo vision, pattern recognition, and motion analysis. Untortunately, existing techniques for image registration tend to be costly.

2. The registration problem

The translational image registration problem can be characterized as follows: We are given functions $F(x)$ and $G(x)$ which give the respective pixel values at each location x in two images, where x is a vector. We wish to find the disparity vector h which minimizes some measure of the difference between $F(x + h)$ and $G(x)$, for x in some region of interest R. (See figure 1).

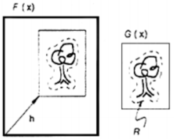
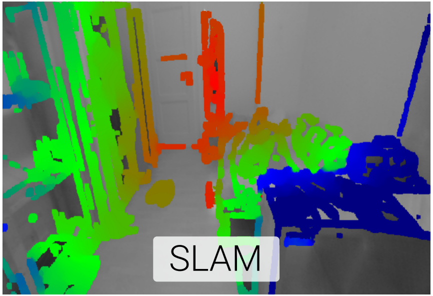


Figure 1: The image registration problem

Applications of Image Registration



Lucas and Kanade: An Iterative Image Registration Technique with an Application to Stereo Vision. IJCAI, 1981.

Lucas-Kanade Algorithm

Objective: Minimize photometric error between template $\mathbf{T}$ and image $\mathbf{I}$

$$\min_{\boldsymbol{\xi}} \|\mathbf{I}(\boldsymbol{\xi}) - \mathbf{T}(\mathbf{0})\|_2^2$$

- $\mathbf{I}(\boldsymbol{\xi})$: image $\mathbf{I}$ transformed by warp parameters $\boldsymbol{\xi}$
- $\mathbf{T}(\mathbf{0})$: template
- **Note:** This is a non-linear objective

Lucas and Kanade: An Iterative Image Registration Technique with an Application to Stereo Vision. IJCAI, 1981.

Lucas-Kanade Algorithm

- **Iteratively** solve the task

$$\boldsymbol{\xi}_{k+1} = \boldsymbol{\xi}_k \circ \Delta\boldsymbol{\xi}$$

Lucas and Kanade: An Iterative Image Registration Technique with an Application to Stereo Vision. IJCAI, 1981.

Lucas-Kanade Algorithm

- **Iteratively** solve the task

$$\boldsymbol{\xi}_{k+1} = \boldsymbol{\xi}_k \circ \Delta\boldsymbol{\xi}$$

- The warp increment $\Delta\boldsymbol{\xi}$ is obtained by **linearizing** the objective

$$\min_{\Delta\boldsymbol{\xi}} \|\mathbf{I}(\boldsymbol{\xi}_k + \Delta\boldsymbol{\xi}) - \mathbf{T}(\mathbf{0})\|_2^2$$

  using first-order **Taylor expansion:**

$$\min_{\Delta\boldsymbol{\xi}} \left\| \mathbf{I}(\boldsymbol{\xi}_k) + \frac{\partial \mathbf{I}(\boldsymbol{\xi}_k)}{\partial \boldsymbol{\xi}} \Delta\boldsymbol{\xi} - \mathbf{T}(\mathbf{0}) \right\|_2^2$$

Lucas and Kanade: An Iterative Image Registration Technique with an Application to Stereo Vision. IJCAI, 1981.

Lucas-Kanade Algorithm

- **Iteratively** solve the task

$$\boldsymbol{\xi}_{k+1} = \boldsymbol{\xi}_k \circ \Delta\boldsymbol{\xi}$$

- The warp increment $\Delta\boldsymbol{\xi}$ is obtained by **linearizing** the objective

$$\min_{\Delta\boldsymbol{\xi}} \|\mathbf{I}(\boldsymbol{\xi}_k + \Delta\boldsymbol{\xi}) - \mathbf{T}(\mathbf{0})\|_2^2$$

  using first-order **Taylor expansion:**

$$\min_{\Delta\boldsymbol{\xi}} \left\| \mathbf{I}(\boldsymbol{\xi}_k) + \frac{\partial \mathbf{I}(\boldsymbol{\xi}_k)}{\partial \boldsymbol{\xi}} \Delta\boldsymbol{\xi} - \mathbf{T}(\mathbf{0}) \right\|_2^2$$

- $\partial \mathbf{I}(\boldsymbol{\xi}_k) / \partial \boldsymbol{\xi}$ must be recomputed at every iteration k

Lucas and Kanade: An Iterative Image Registration Technique with an Application to Stereo Vision. IJCAI, 1981.

Inverse Compositional Algorithm

- **Iteratively** solve the task

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

Baker and Matthews: Lucas-Kanade 20 Years On: A Unifying Framework: Part 1. Technical Report, Carnegie Mellon University, 2003.

Inverse Compositional Algorithm

- **Iteratively** solve the task

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

- The warp increment $\Delta\boldsymbol{\xi}$ is obtained by **linearizing** the objective

$$\min_{\Delta\boldsymbol{\xi}} \|\mathbf{I}(\boldsymbol{\xi}_k) - \mathbf{T}(\Delta\boldsymbol{\xi})\|_2^2$$

using first-order **Taylor expansion:**

$$\min_{\Delta\boldsymbol{\xi}} \left\| \mathbf{I}(\boldsymbol{\xi}_k) - \mathbf{T}(\mathbf{0}) - \frac{\partial \mathbf{T}(\mathbf{0})}{\partial \boldsymbol{\xi}} \Delta\boldsymbol{\xi} \right\|_2^2$$

Inverse Compositional Algorithm

- **Iteratively** solve the task

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

- The warp increment $\Delta\boldsymbol{\xi}$ is obtained by **linearizing** the objective

$$\min_{\Delta\boldsymbol{\xi}} \|\mathbf{I}(\boldsymbol{\xi}_k) - \mathbf{T}(\Delta\boldsymbol{\xi})\|_2^2$$

  using first-order **Taylor expansion:**

$$\min_{\Delta\boldsymbol{\xi}} \left\| \mathbf{I}(\boldsymbol{\xi}_k) - \mathbf{T}(\mathbf{0}) - \frac{\partial \mathbf{T}(\mathbf{0})}{\partial \boldsymbol{\xi}} \Delta\boldsymbol{\xi} \right\|_2^2$$

- $\partial \mathbf{T}(\mathbf{0}) / \partial \boldsymbol{\xi}$ does not depend on $\boldsymbol{\xi}_k$ and can thus be pre-computed

Baker and Matthews: Lucas-Kanade 20 Years On: A Unifying Framework: Part 1. Technical Report, Carnegie Mellon University, 2003.

Comparison

Lucas-Kanade Algorithm

$$\boldsymbol{\xi}_{k+1} = \boldsymbol{\xi}_k \circ \Delta\boldsymbol{\xi}$$

$$\min_{\Delta\boldsymbol{\xi}} \|\mathbf{I}(\boldsymbol{\xi}_k + \Delta\boldsymbol{\xi}) - \mathbf{T}(\mathbf{0})\|_2^2$$

$$\min_{\Delta\boldsymbol{\xi}} \left\| \mathbf{I}(\boldsymbol{\xi}_k) + \frac{\partial \mathbf{I}(\boldsymbol{\xi}_k)}{\partial \boldsymbol{\xi}} \Delta\boldsymbol{\xi} - \mathbf{T}(\mathbf{0}) \right\|_2^2$$

Inverse Compositional Algorithm

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

$$\min_{\Delta\boldsymbol{\xi}} \|\mathbf{I}(\boldsymbol{\xi}_k) - \mathbf{T}(\Delta\boldsymbol{\xi})\|_2^2$$

$$\min_{\Delta\boldsymbol{\xi}} \left\| \mathbf{I}(\boldsymbol{\xi}_k) - \mathbf{T}(\mathbf{0}) - \frac{\partial \mathbf{T}(\mathbf{0})}{\partial \boldsymbol{\xi}} \Delta\boldsymbol{\xi} \right\|_2^2$$

- The Inverse Compositional Algorithm is more **computationally efficient**!

Baker and Matthews: Lucas-Kanade 20 Years On: A Unifying Framework: Part 1. Technical Report, Carnegie Mellon University, 2003.

Robust M-Estimation

- To handle outliers, **robust estimation** can be used:

$$\min_{\Delta\boldsymbol{\xi}} \ \mathbf{r}_k(\Delta\boldsymbol{\xi})^T \, \mathbf{W} \, \mathbf{r}_k(\Delta\boldsymbol{\xi})$$

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

- The diagonal weight matrix $\mathbf{W}$ is determined by the **implicit robust loss** $\rho(\cdot)$

Baker, Gross, Matthews and Ishikawa: Lucas-Kanade 20 Years On: A Unifying Framework: Part 2. Technical Report, Carnegie Mellon University, 2003.

Optimization

- The minimizer of $\mathbf{r}_k(\Delta\boldsymbol{\xi})^T\,\mathbf{W}\,\mathbf{r}_k(\Delta\boldsymbol{\xi})$ leads to the **Gauss-Newton update**:

$$\Delta\boldsymbol{\xi} = \left(\mathbf{J}^T\mathbf{W}\mathbf{J}\right)^{-1}\mathbf{J}^T\mathbf{W}\,\mathbf{r}_k(\mathbf{0})$$

with Jacobian $\mathbf{J} = \partial\mathbf{T}(\mathbf{0})/\partial\boldsymbol{\xi}$

Baker, Gross, Matthews and Ishikawa: Lucas-Kanade 20 Years On: A Unifying Framework: Part 2. Technical Report, Carnegie Mellon University, 2003.

Optimization

- The minimizer of $\mathbf{r}_k(\Delta\boldsymbol{\xi})^T\,\mathbf{W}\,\mathbf{r}_k(\Delta\boldsymbol{\xi})$ leads to the **Gauss-Newton update**:

$$\Delta\boldsymbol{\xi} = \left(\mathbf{J}^T\mathbf{W}\mathbf{J}\right)^{-1}\mathbf{J}^T\mathbf{W}\,\mathbf{r}_k(\mathbf{0})$$

  with Jacobian $\mathbf{J} = \partial\mathbf{T}(\mathbf{0})/\partial\boldsymbol{\xi}$

- As the approximate Hessian $\mathbf{J}^T\mathbf{W}\mathbf{J}$ easily becomes ill-conditioned, a **damping term** is added in practice, resulting in a **trust-region** update:

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

Baker, Gross, Matthews and Ishikawa: Lucas-Kanade 20 Years On: A Unifying Framework: Part 2. Technical Report, Carnegie Mellon University, 2003.

Optimization

- The minimizer of $\mathbf{r}_k(\Delta\boldsymbol{\xi})^T\,\mathbf{W}\,\mathbf{r}_k(\Delta\boldsymbol{\xi})$ leads to the **Gauss-Newton update**:

$$\Delta\boldsymbol{\xi} = \left(\mathbf{J}^T\mathbf{W}\mathbf{J}\right)^{-1}\mathbf{J}^T\mathbf{W}\,\mathbf{r}_k(\mathbf{0})$$

  with Jacobian $\mathbf{J} = \partial\mathbf{T}(\mathbf{0})/\partial\boldsymbol{\xi}$
- As the approximate Hessian $\mathbf{J}^T\mathbf{W}\mathbf{J}$ easily becomes ill-conditioned, a **damping term** is added in practice, resulting in a **trust-region** update:

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

- For different λ, the update varies between the **Gauss-Newton** direction and **steepest descent**. In practice, λ is chosen based on simple **heuristics**.

Baker, Gross, Matthews and Ishikawa: Lucas-Kanade 20 Years On: A Unifying Framework: Part 2. Technical Report, Carnegie Mellon University, 2003.

Robust Inverse Compositional Algorithm

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

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

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

Baker, Gross, Matthews and Ishikawa: Lucas-Kanade 20 Years On: A Unifying Framework: Part 2. Technical Report, Carnegie Mellon University, 2003.

What is the problem?

Limitations:

- Easily gets trapped in **local minima** as residuals often highly non-linear

What is the problem?

Limitations:

- Easily gets trapped in **local minima** as residuals often highly non-linear
- Choosing a **robust loss** function ρ is difficult as residual distribution unknown

Baker, Gross, Matthews and Ishikawa: Lucas-Kanade 20 Years On: A Unifying Framework: Part 2. Technical Report, Carnegie Mellon University, 2003.

What is the problem?

Limitations:

- Easily gets trapped in **local minima** as residuals often highly non-linear
- Choosing a **robust loss** function ρ is difficult as residual distribution unknown
- The objective does not capture **high-order statistics** of the inputs ($\mathbf{W}$ is diagonal)

Baker, Gross, Matthews and Ishikawa: Lucas-Kanade 20 Years On: A Unifying Framework: Part 2. Technical Report, Carnegie Mellon University, 2003.

What is the problem?

Limitations:

- Easily gets trapped in **local minima** as residuals often highly non-linear
- Choosing a **robust loss** function ρ is difficult as residual distribution unknown
- The objective does not capture **high-order statistics** of the inputs ($\mathbf{W}$ is diagonal)
- **Damping heuristics** are suboptimal and do not depend on the input

Baker, Gross, Matthews and Ishikawa: Lucas-Kanade 20 Years On: A Unifying Framework: Part 2. Technical Report, Carnegie Mellon University, 2003.

What is the problem?

Limitations:

- Easily gets trapped in **local minima** as residuals often highly non-linear
- Choosing a **robust loss** function ρ is difficult as residual distribution unknown
- The objective does not capture **high-order statistics** of the inputs ($\mathbf{W}$ is diagonal)
- **Damping heuristics** are suboptimal and do not depend on the input

Our Approach

- Unroll the algorithm into a parameterized **feed-forward network**

Baker, Gross, Matthews and Ishikawa: Lucas-Kanade 20 Years On: A Unifying Framework: Part 2. Technical Report, Carnegie Mellon University, 2003.

What is the problem?

Limitations:

- Easily gets trapped in **local minima** as residuals often highly non-linear
- Choosing a **robust loss** function ρ is difficult as residual distribution unknown
- The objective does not capture **high-order statistics** of the inputs ($\mathbf{W}$ is diagonal)
- **Damping heuristics** are suboptimal and do not depend on the input

Our Approach

- Unroll the algorithm into a parameterized **feed-forward network**
- **Relax assumptions** above but preserves advantages of robust estimation

Baker, Gross, Matthews and Ishikawa: Lucas-Kanade 20 Years On: A Unifying Framework: Part 2. Technical Report, Carnegie Mellon University, 2003.

What is the problem?

Limitations:

- Easily gets trapped in **local minima** as residuals often highly non-linear
- Choosing a **robust loss** function ρ is difficult as residual distribution unknown
- The objective does not capture **high-order statistics** of the inputs ($\mathbf{W}$ is diagonal)
- **Damping heuristics** are suboptimal and do not depend on the input

Our Approach

- Unroll the algorithm into a parameterized **feed-forward network**
- **Relax assumptions** above but preserves advantages of robust estimation
- Trained **end-to-end** from data

Baker, Gross, Matthews and Ishikawa: Lucas-Kanade 20 Years On: A Unifying Framework: Part 2. Technical Report, Carnegie Mellon University, 2003.

Approach

Robust Inverse Compositional Algorithm

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

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

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

Robust Inverse Compositional Algorithm

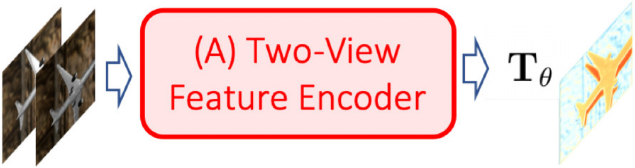
$$\mathbf{r}_k = \boxed{\mathbf{I}_\theta(\boldsymbol{\xi}_k)} - \boxed{\mathbf{T}_\theta(\mathbf{0})}$$

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

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

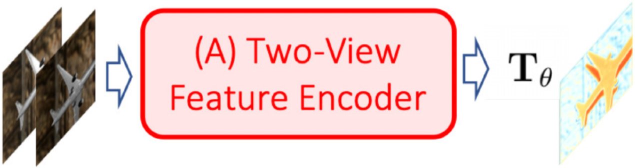
- Two-view feature encoder   Convolutional m-estimator   Trust-region network

Two-View Feature Encoder



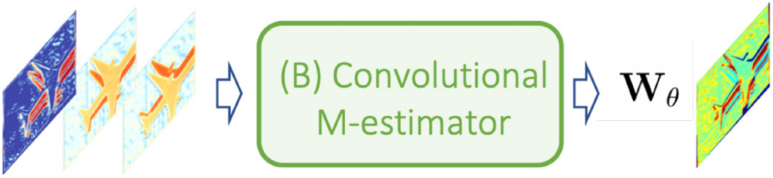
- **ConvNet** ϕ_θ for extracting:
  - Image features $\mathbf{I}_\theta = \phi_\theta([\mathbf{I}, \mathbf{T}])$
  - Template features $\mathbf{T}_\theta = \phi_\theta([\mathbf{T}, \mathbf{I}])$

Two-View Feature Encoder



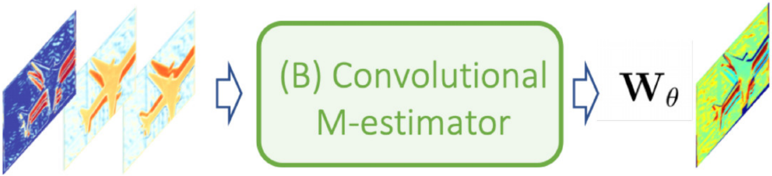
- **ConvNet** ϕ_θ for extracting:
  - Image features $\mathbf{I}_\theta = \phi_\theta([\mathbf{I}, \mathbf{T}])$
  - Template features $\mathbf{T}_\theta = \phi_\theta([\mathbf{T}, \mathbf{I}])$
- Both views passed as input
- Features capture **high-order spatial and temporal information**

Convolutional M-Estimator



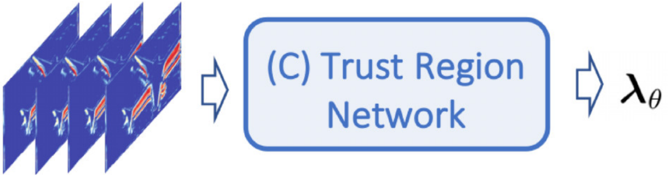
- **Robust weight function** parameterized by **ConvNet** ψ_θ
  - Input: feature maps $\mathbf{I}, \mathbf{T}$ and residual $\mathbf{r}$
  - Output: diagonal weight matrix $\mathbf{W}_\theta = \psi_\theta(\mathbf{I}, \mathbf{T}, \mathbf{r})$

Convolutional M-Estimator



- **Robust weight function** parameterized by **ConvNet** ψ_θ
  - Input: feature maps $\mathbf{I}, \mathbf{T}$ and residual $\mathbf{r}$
  - Output: diagonal weight matrix $\mathbf{W}_\theta = \psi_\theta(\mathbf{I}, \mathbf{T}, \mathbf{r})$
- Robust function is **learned end-to-end** from data
- Robust function **conditioned** on input image/template and pixel context

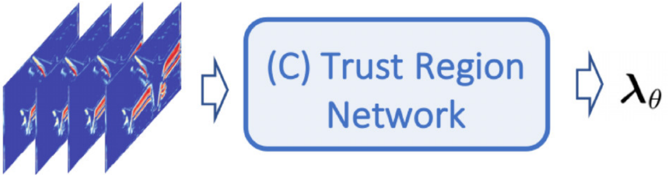
Trust Region Network



- Compute hypothetical updates for a set of **damping proposals:**

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

Trust Region Network



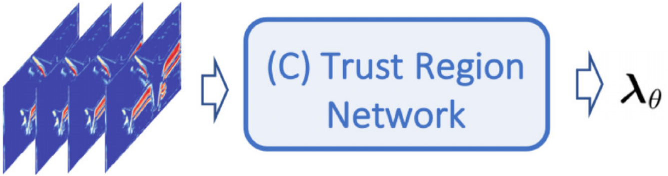
- Compute hypothetical updates for a set of **damping proposals:**

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

- Pass resulting **residuals** to a neural net which predicts **damping parameters:**

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

Trust Region Network



- Compute hypothetical updates for a set of **damping proposals:**

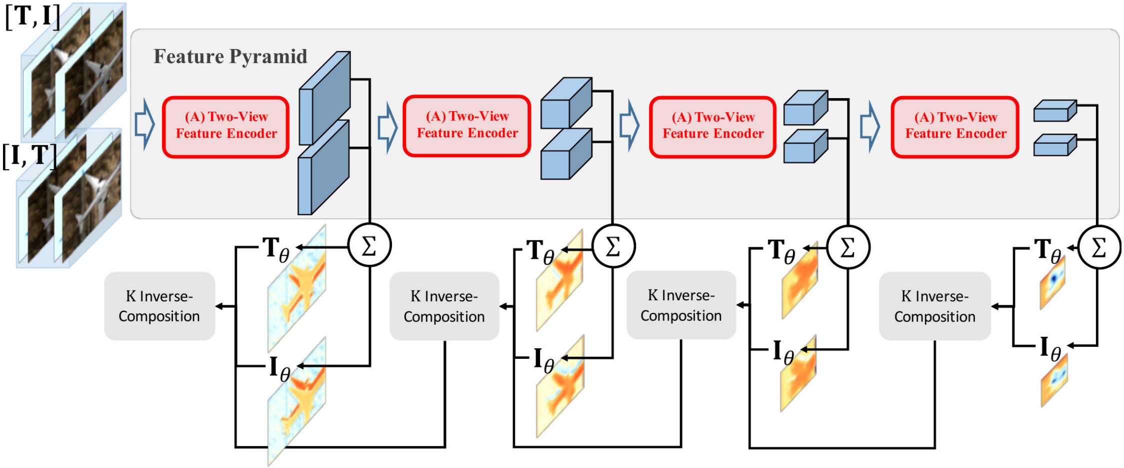
$$\Delta\boldsymbol{\xi}_i = \left(\mathbf{J}^T\mathbf{W}\mathbf{J} + \lambda_i \,\mathrm{diag}(\mathbf{J}^T\mathbf{W}\mathbf{J})\right)^{-1}\mathbf{J}^T\mathbf{W}\,\mathbf{r}_k(\mathbf{0})$$

- Pass resulting **residuals** to a neural net which predicts **damping parameters:**

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

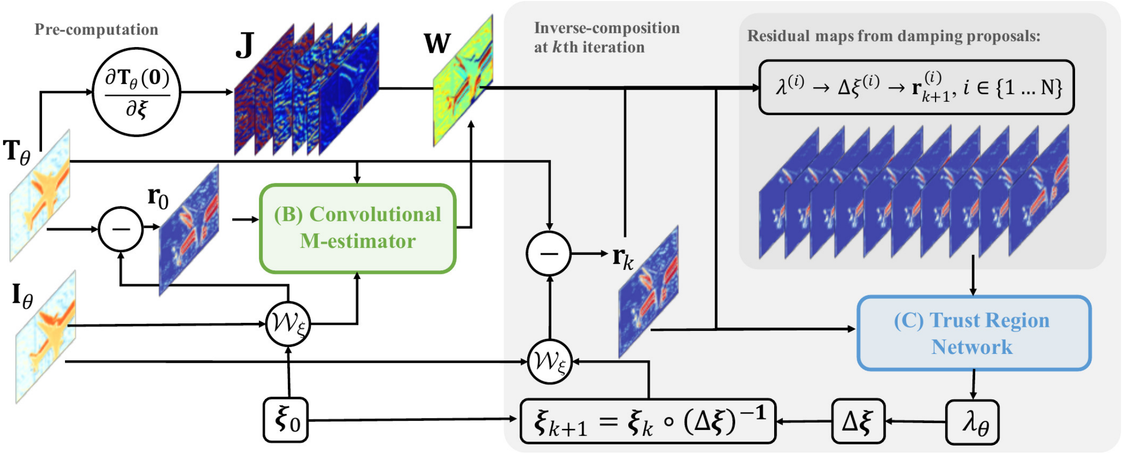
- Our experiments show that residual maps indeed contain valuable information

Overview



[T, I]
[I, T]
Feature Pyramid
(A) Two-View Feature Encoder
(A) Two-View Feature Encoder
(A) Two-View Feature Encoder
(A) Two-View Feature Encoder
Σ
Σ
Σ
Σ
$\mathbf{T}_\theta$
$\mathbf{I}_\theta$
$\mathbf{T}_\theta$
$\mathbf{I}_\theta$
$\mathbf{T}_\theta$
$\mathbf{I}_\theta$
$\mathbf{T}_\theta$
$\mathbf{I}_\theta$
K Inverse-Composition
K Inverse-Composition
K Inverse-Composition
K Inverse-Composition

Overview



Pre-computation
$\frac{\partial \mathbf{T}_\theta(\mathbf{0})}{\partial \xi}$
J
W
Inverse-composition at kth iteration
Residual maps from damping proposals:
$\lambda^{(i)} \rightarrow \Delta\xi^{(i)} \rightarrow \mathbf{r}_{k+1}^{(i)}, i \in \{1 \ldots \mathrm{N}\}$
$\mathbf{T}_\theta$
$\mathbf{r}_0$
(B) Convolutional M-estimator
$\mathbf{r}_k$
$\mathbf{I}_\theta$
$\mathcal{W}_\xi$
(C) Trust Region Network
ξ_0
$\xi_{k+1} = \xi_k \circ (\Delta\xi)^{-1}$
$\Delta\xi$
λ_θ

Experimental Evaluation

RGB-D Image Alignment

The rigid body transformation $\mathbf{T}_{\boldsymbol{\xi}}$ warps pixel $\mathbf{x}$ as

$$\mathcal{W}_{\boldsymbol{\xi}}(\mathbf{x}) = \mathbf{K}\,\mathbf{T}_{\boldsymbol{\xi}}\,D(\mathbf{x})\mathbf{K}^{-1}\,\mathbf{x}$$

with

- $\mathbf{K}$: camera intrinsics $D(\mathbf{x})$: depth at pixel $\mathbf{x}$
- $\mathbf{I}_{\theta}(\boldsymbol{\xi})$ is obtained via bilinear sampling with z-buffering

Training Objective

3D End-Point-Error Loss:

$$\mathcal{L} = \frac{1}{|\mathcal{P}|} \sum_{l \in \mathcal{L}} \sum_{\mathbf{p} \in \mathcal{P}} \|\mathbf{T}_{gt}\,\mathbf{p} - \mathbf{T}(\boldsymbol{\xi}_l)\,\mathbf{p}\|_2^2$$

with

- $\mathbf{p} = D(\mathbf{x})\mathbf{K}^{-1}\mathbf{x}$: 3D point corresponding to pixel $\mathbf{x}$ in $\mathbf{I}$
- $\mathcal{L}$: set of coarse-to-fine pyramid levels

The EPE loss balances the influences of translation and rotation.

Datasets

Object Motion:

- MovingObjects3D (ShapeNet objects moving in static 3D room)

Datasets

Object Motion:

- ► MovingObjects3D (ShapeNet objects moving in static 3D room)

Camera Motion:

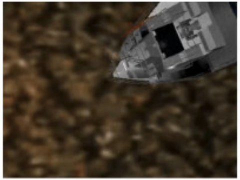
- ► BundleFusion [Dai et al., 2017]
- ► DynamicBundleFusion [Lv et al., 2018]
- ► TUM RGB-D SLAM [Sturm et al., 2012]

We subsample frames to increase the motion/difficulty.

Datasets

Train objects

Test objects



Baselines

Classical Methods:

- ► ICP implementation of Open3D [Zhou et al., 2018]
- ► RGB-D Visual Odometry [Steinbrücker et al., 2011]

Baselines

Classical Methods:

- ICP implementation of Open3D [Zhou et al., 2018]
- RGB-D Visual Odometry [Steinbrücker et al., 2011]

Direct Pose Regression:

- Pose Regression with a FlowNetSimple backbone [Dosovitskiy et al., 2015]
- Cascaded Pose Regression
- Pose Regression with IC Refinement [Li et al., 2018]

Baselines

Classical Methods:

- ► ICP implementation of Open3D [Zhou et al., 2018]
- ► RGB-D Visual Odometry [Steinbrücker et al., 2011]

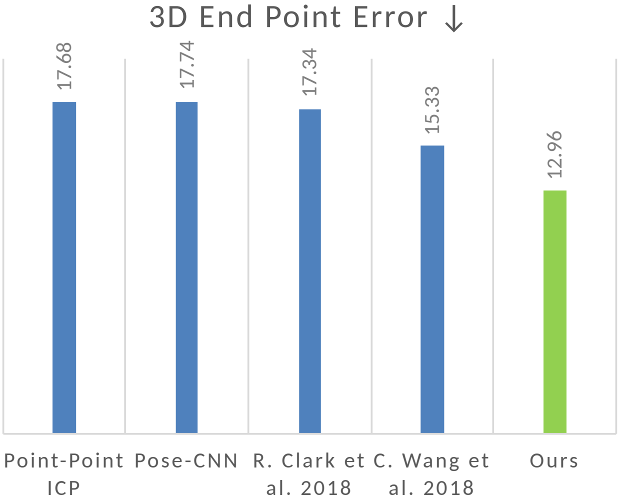
Direct Pose Regression:

- ► Pose Regression with a FlowNetSimple backbone [Dosovitskiy et al., 2015]
- ► Cascaded Pose Regression
- ► Pose Regression with IC Refinement [Li et al., 2018]

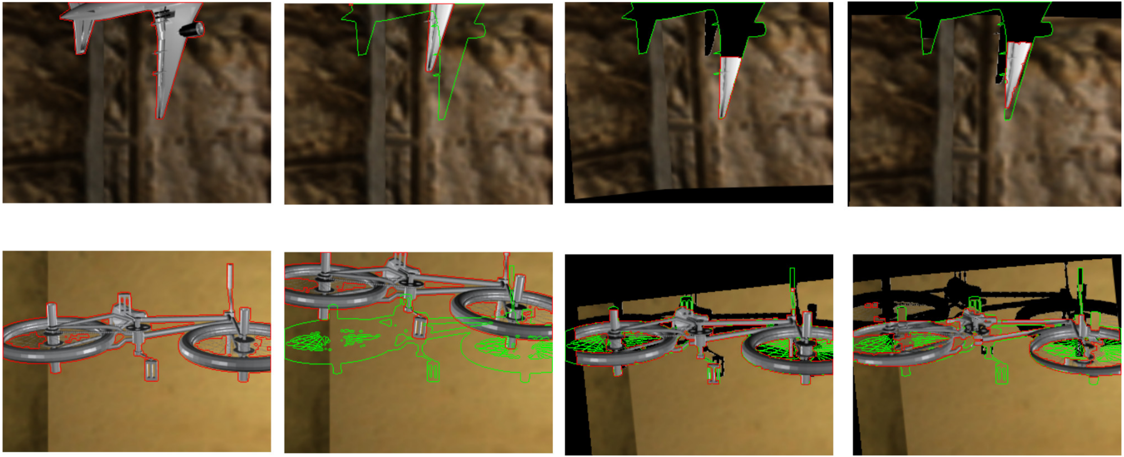
Learning-based Optimization:

- ► LS-Net [Clark et al., 2018]
- ► DeepLK [Wang et al., 2018]

Results on MovingObjects3D

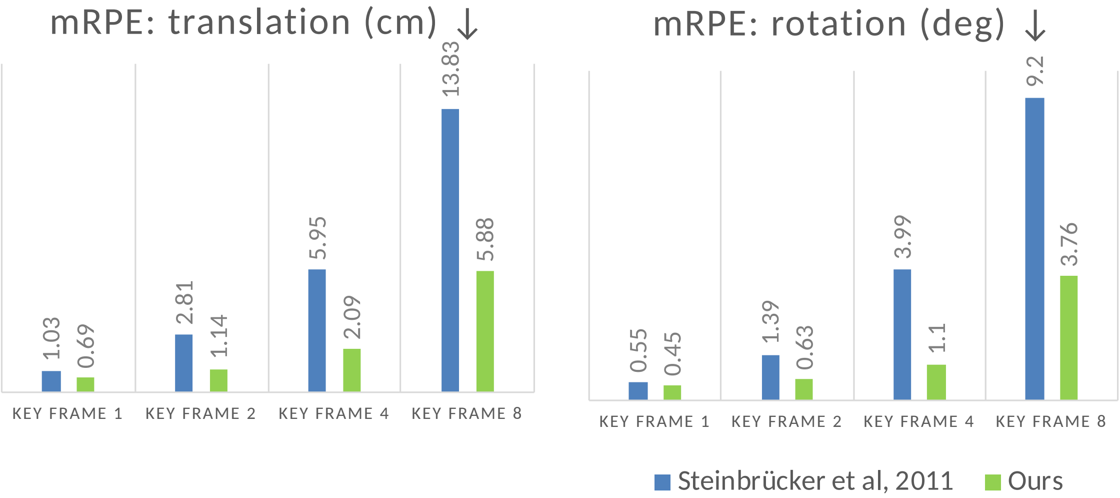


Results on MovingObjects3D



$\mathbf{T}$ $\quad$ $\mathbf{I}$ $\quad$ $\mathbf{I}(\boldsymbol{\xi}^{\mathrm{GT}})$ $\quad$ $\mathbf{I}(\xi^{*})$

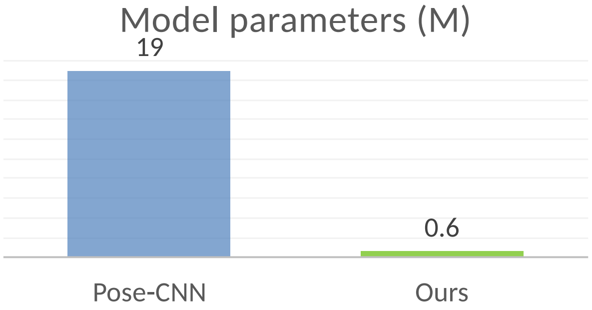
Results on TUM RGB-D

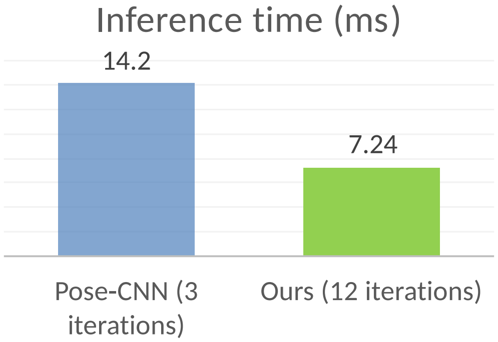


Ablation Study on DynamicBundleFusion

Method	3D EPE (cm)
No learning	8.58
Ours (A)	7.11
Ours (A)+(B)	6.88
Ours (A)+(B)+(C)	4.64
Ours (A)+(B)+(C) (no WS)	**3.82**

Model Parameters and Inference Time





Summary

- **Generalization** of **Lucas-Kanade** algorithm lifting several **assumptions**

Summary

- **Generalization** of **Lucas-Kanade** algorithm lifting several **assumptions**
- **3 modules:**
  - Two-view Feature Encoder
  - Convolutional M-Estimator
  - Trust Region Network

Summary

- **Generalization** of **Lucas-Kanade** algorithm lifting several **assumptions**
- **3 modules:**
  - Two-view Feature Encoder
  - Convolutional M-Estimator
  - Trust Region Network
- **End-to-end** trainable

Summary

- **Generalization** of **Lucas-Kanade** algorithm lifting several **assumptions**
- **3 modules:**
  - Two-view Feature Encoder
  - Convolutional M-Estimator
  - Trust Region Network
- **End-to-end** trainable
- Evaluated on object motion and camera motion estimation tasks

Summary

- **Generalization** of **Lucas-Kanade** algorithm lifting several **assumptions**
- **3 modules:**
  - Two-view Feature Encoder
  - Convolutional M-Estimator
  - Trust Region Network
- **End-to-end** trainable
- Evaluated on object motion and camera motion estimation tasks
- Better **generalization** than image-to-pose regression models

Summary

- **Generalization** of **Lucas-Kanade** algorithm lifting several **assumptions**
- **3 modules:**
  - Two-view Feature Encoder
  - Convolutional M-Estimator
  - Trust Region Network
- **End-to-end** trainable
- Evaluated on object motion and camera motion estimation tasks
- Better **generalization** than image-to-pose regression models
- Higher **accuracy** compared to classical (non-learned) models

Summary

- **Generalization** of **Lucas-Kanade** algorithm lifting several **assumptions**
- **3 modules:**
  - Two-view Feature Encoder
  - Convolutional M-Estimator
  - Trust Region Network
- **End-to-end** trainable
- Evaluated on object motion and camera motion estimation tasks
- Better **generalization** than image-to-pose regression models
- Higher **accuracy** compared to classical (non-learned) models

Conclusion: Combining classical and deep methods increases robustness

Thank you!

http://autonomousvision.github.io