LIFT: Learned Invariant Feature Transform

Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, Pascal Fua
Poster: S-4A-08
Feature point pipeline

**Keypoint Detection**

**Orientation Estimation**

**Feature Description**
- Heuristics:
  - Dominant Gradient Orientations (SIFT, SURF,...)
  - Center of Mass (ORB,...)
Enhanced performance with Deep Learning

Avg. matching score on ‘Strecha’

Avg. matching score on ‘DTU’

Avg. matching score on ‘Webcam’

LIFT: Learned Invariant Feature Transform
Enhanced performance with Deep Learning

- LIFT: Learned Invariant Feature Transform

Comparison of matching scores across datasets ('Strecha', 'DTU', 'Webcam') for various feature descriptors.

- SIFT, SURF, ORB, Daisy, sGLOH, MROGH, LIOP, BiCE
- BRISK, FREAK, VGG, DeepDesc, PN-Net, KAZE, LIFT

Scores indicate relative performance, with higher values indicating better matching performance.
Learned Local Features

KEYPOINT DETECTION

ORIENTATION ESTIMATION

FEATURE DESCRIPTION
Learned Local Features

LIFT Network

- **Drop-in** replacement for SIFT
- **Practical** runtime (1.5x ~ 3x SIFT)


"Learning to Assign Orientations to Feature Points", CVPR 2016.


LIFT: Learned Invariant Feature Transform
Matching features on ‘Strecha’, sequence ‘Fountain’.
Correct matches shown with green lines.

**SIFT.** Average: **60.2** matches

Matches: 20 / 500

**LIFT (Ours).** Average: **98.6** matches

Matches: 48 / 500

+64%
Matching features on Webcam sequence ‘Frankfurt’.
Correct matches depicted by green lines.

**SIFT.** Average: 23.1 matches

**LIFT (Ours).** Average: 60.6 matches

Matches: 2 / 500

Matches: 7 / 500

+162%
The LIFT Network

Patch based pipeline for scalable learning
(at training time only)
The LIFT Network


“Glue”, to preserve differentiability:
- Spatial Transformer Networks, NIPS 2015.


The LIFT Network


"Glue", to preserve differentiability:
- Spatial Transformer Networks, NIPS 2015.


Training with patches from SfM

- We generate training pairs with **Structure from Motion (SfM)**.
- We use **two photo-tourism sets**, for robustness to imaging changes.
Training requires various patches

Keypoint VS Non-Keypoint

Matching Keypoints

Matching Keypoints VS Non-Matching Keypoints
Quadruplet Siamese Network

P^1 → DET → softmax → x^1 → Crop → p^1 → ORI → θ^1 → Rot → p^θ_1 → DESC → d^1

P^2 → DET → softmax → x^2 → Crop → p^2 → ORI → θ^2 → Rot → p^θ_2 → DESC → d^2

P^3 → DET → softmax → x^3 → Crop → p^3 → ORI → θ^3 → Rot → p^θ_3 → DESC → d^3

P^4 → DET
A single, global cost function

\[
\min_{\{f_\mu, g_\phi, h_\rho\}} \sum_{(P_1, P_2, P_3, P_4)} \gamma \mathcal{L}_{\text{class}}(P^1, P^2, P^3, P^4) + \mathcal{L}_{\text{pair}}(P^1, P^2)
\]

\[
\mathcal{L}_{\text{class}}(P^1, P^2, P^3, P^4) = \sum_{i=1}^{4} \alpha_i \max(0, (1 - \text{softmax}(f_\mu(P^i)) y_i))^2
\]

\[
\mathcal{L}_{\text{pair}}(P^1, P^2) = \| h_\rho(G(P^1, \text{softargmax}(f_\mu(P^1)))) - h_\rho(G(P^2, \text{softargmax}(f_\mu(P^2)))) \|_2
\]

\[
G(P, x) = \text{Rot}(P, x, g_\phi(\text{Crop}(P, x)))
\]

LIFT: Learned Invariant Feature Transform
Significant improvement over state-of-the-art

LIFT with ‘pic’ dataset

LIFT with ‘rf’ dataset

Avg. matching score on ‘Strecha’

Avg. matching score on ‘DTU’

Avg. matching score on ‘Webcam’

LIFT: Learned Invariant Feature Transform
Each component is *meant for* each other.

Descriptor performance (NN mAP)

- **SIFT descriptor when used with SIFT keypoints**
- **LIFT descriptor when used with LIFT keypoints**
Each component is *meant for* each other

LIFT descriptor performs best with LIFT keypoints

Descriptor performance (NN mAP)
Practical runtime

SCALE-SPACE IMAGE → DET → SCORE PYRAMID → NMS → KEYPOINTS

Crop → ORI → Rot → DESC

d_1
d_2
...
d_N
Practical runtime

Can be decoupled for efficient evaluation
Summary

• **LIFT**: Learned Invariant Feature Transform

• Practical, **drop-in** replacement for SIFT

• **Efficient** and **outperforms** the state-of-the-art

• **Code** available at [https://github.com/cvlab-epfl/LIFT](https://github.com/cvlab-epfl/LIFT)

• Poster S-4A-08