Supplementary Appendix for LIFT: Learned Invariant Feature Transform

Kwang Moo Yi*,[1], Eduard Trulls*,[1], Vincent Lepetit[2], Pascal Fua[1]

[1]Computer Vision Laboratory, Ecole Polytechnique Fédérale de Lausanne (EPFL)
[2]Institute for Computer Graphics and Vision, Graz University of Technology
{kwang.yi, eduard.trulls, pascal.fua}@epfl.ch, lepetit@icg.tugraz.at

1 Supplementary Appendix

In this appendix, we provide the implementation details necessary to make our experimental results reproducible. Second, we provide references to the algorithms used as baselines.

1.1 Implementation details

Table 1 details the parameters of the different components in our LIFT network.

<table>
<tr><th></th><th>Conv. filter</th><th># of nodes</th><th>Pooling</th><th>Activation</th><th>Stride</th></tr>
<tr><td>Kp-1</td><td>25 × 25</td><td>1</td><td>—</td><td>GHH, $N = 4$, $M = 4$</td><td>—</td></tr>
<tr><td>Kp-2</td><td colspan="5">softargmax Mapping</td></tr>
<tr><td>Kp-3</td><td colspan="5">Spatial Transformer ‘Crop’</td></tr>
<tr><td>Ori-1</td><td>5 × 5</td><td>10</td><td>Max pooling</td><td>ReLU</td><td>—</td></tr>
<tr><td>Ori-2</td><td>5 × 5</td><td>20</td><td>Max pooling</td><td>ReLU</td><td>—</td></tr>
<tr><td>Ori-3</td><td>3 × 3</td><td>50</td><td>Max pooling</td><td>ReLU</td><td>—</td></tr>
<tr><td>Ori-4</td><td>1 × 1</td><td>100</td><td>—</td><td>GHH, $N = 4$, $M = 4$</td><td>—</td></tr>
<tr><td>Ori-5</td><td>1 × 1</td><td>2</td><td>—</td><td>GHH, $N = 4$, $M = 4$</td><td>—</td></tr>
<tr><td>Ori-6</td><td colspan="5">arctan Mapping</td></tr>
<tr><td>Ori-7</td><td colspan="5">Spatial Transformer ‘Rot’</td></tr>
<tr><td>Desc-1</td><td>7 × 7</td><td>32</td><td>l_2 pooling, 2 × 2</td><td>Tanh</td><td>2</td></tr>
<tr><td>Desc-2</td><td>6 × 6</td><td>64</td><td>l_2 pooling, 3 × 3</td><td>Tanh</td><td>3</td></tr>
<tr><td>Desc-3</td><td>5 × 5</td><td>128</td><td>l_2 pooling, 4 × 4</td><td>Tanh</td><td>4</td></tr>
</table>

Table 1. The LIFT network architecture.

During the descriptor training, we use Stochastic Gradient Descent with momentum. With the trained descriptor, we train the Orientation Estimator with the ADAM [1] solver. Finally, the Detector is trained in conjunction with the two components using ADAM as well.

$*$ First two authors contributed equally.

This work was supported in part by the EU FP7 project MAGELLAN under grant number ICT-FP7-611526.

1.2 Implementations of the Compared Methods

For fair comparison, we used the authors implementations where applicable, as well as the publicly available ones. All methods were used with default parameters, unless stated otherwise.

- SIFT [2]: OpenCV library.
  `http://opencv.org/downloads.html`
- SIFT-HesAff [3]: VLFeat library.
  `http://www.vlfeat.org/download.html`
- SURF [4]: OpenCV library.
  `http://opencv.org/downloads.html`
- KAZE [5]: Implementation by the authors.
  `https://github.com/pablofdezalc/kaze`
- ORB [6]: OpenCV library.
  `http://opencv.org/downloads.html`
  To obtain enough keypoints we used nFeatures=1000 and nLevels=3.
- Daisy [7]: Provided by the authors.
  `https://github.com/etola/libdaisy`
  Patches were extracted to be four times the scale, which was the value authors used in [8].
- sGLOH [9]: Implementation by the authors.
  `http://www.math.unipa.it/fbellavia/htm/research.html`
- MROGH [10]: Implementation by the authors.
  `https://github.com/bfan/MROGH-feature-descriptor`
- Edge Foci [8] and BiCE [7]: Implementation by the authors.
  `http://research.microsoft.com/en-us/downloads/ae6ab93a-4e17-46b9-a1e4-907aeb64d1e5/`
- BRISK [11]: Implementation by the authors.
  `http://www.asl.ethz.ch/people/lestefan/personal/BRISK`
  To obtain enough keypoints we used threshold of 20.
- FREAK [12]: Implementation by the authors.
  `https://github.com/kikohs/freak`
- VGG [13]: Implementation by the authors.
  `http://www.robots.ox.ac.uk/~vgg/software/learn_desc/`
  `http://www.vlfeat.org/download.html`
  Patches were extracted with the VLFeat library, with a relativeExtent of 7.5, which is the same as what SIFT uses. We use the pre-learned model learned with the *liberty* dataset from [13].
- MatchNet [14]: Implementation by the authors.
  `https://github.com/hanxf/matchnet`
  We use the models learned with the *liberty* dataset.
- DeepDesc [15]: Implementation by the authors.
  `https://github.com/etrulls/deepdesc-release`
  We used the models trained on split #4.
- PN-Net [16]: Implementation by the authors.
  `https://github.com/vbalnt/pnnet`
  We used the networks trained on the *liberty* dataset.

References

1. Kingma, D., Ba, J.: Adam: A Method for Stochastic Optimisation. In: ICLR. (2015)
2. Lowe, D.: Distinctive Image Features from Scale-Invariant Keypoints. IJCV **20**(2) (2004)
3. Mikolajczyk, K., Schmid, C.: An Affine Invariant Interest Point Detector. In: ECCV. (2002) 128–142
4. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded Up Robust Features. In: ECCV. (2006)
5. Alcantarilla, P., Fernández, P., Bartoli, A., Davidson, A.J.: KAZE Features. In: ECCV. (2012)
6. Rublee, E., Rabaud, V., Konolidge, K., Bradski, G.: ORB: An Efficient Alternative to SIFT or SURF. In: ICCV. (2011)
7. Zitnick, C.: Binary Coherent Edge Descriptors. In: ECCV. (2010)
8. Zitnick, C., Ramnath, K.: Edge Foci Interest Points. In: ICCV. (2011)
9. Bellavia, F., Tegolo, D.: Improving Sift-Based Descriptors Stability to Rotations. In: ICPR. (2010)
10. Fan, B., Wu, F., Hu, Z.: Aggregating Gradient Distributions into Intensity Orders: A Novel Local Image Descriptor. In: CVPR. (2011)
11. Leutenegger, S., Chli, M., Siegwart, R.: BRISK: Binary Robust Invariant Scalable Keypoints. In: ICCV. (2011)
12. Alahi, A., Ortiz, R., Vandergheynst, P.: FREAK: Fast Retina Keypoint. In: CVPR. (2012)
13. Simonyan, K., Vedaldi, A., Zisserman, A.: Learning Local Feature Descriptors Using Convex Optimisation. PAMI (2014)
14. Han, X., Leung, T., Jia, Y., Sukthankar, R., Berg, A.C.: MatchNet: Unifying Feature and Metric Learning for Patch-Based Matching. In: CVPR. (2015)
15. Simo-Serra, E., Trulls, E., Ferraz, L., Kokkinos, I., Fua, P., Moreno-Noguer, F.: Discriminative Learning of Deep Convolutional Feature Point Descriptors. In: ICCV. (2015)
16. Balntas, V., Johns, E., Tang, L., Mikolajczyk, K.: PN-Net: Conjoined Triple Deep Network for Learning Local Image Descriptors. In: arXiv Preprint. (2016)