24^{th} *Computer Vision Winter Workshop*
Friedrich Fraundorfer, Peter M. Roth, Fabian Schenk (eds.)
Stift Vorau, Austria, February 6–8, 2019

DOI: 10.3217/978-3-85125-652-9-06

Leveraging Outdoor Webcams for Local Descriptor Learning

Milan Pultar, Dmytro Mishkin, Jiří Matas
Visual Recognition Group, Dept. of Cybernetics
Faculty of Electrical Engineering, CTU in Prague
milan.pultar@gmail.com, {mishkdmy, matas}@cmp.felk.cvut.cz

Abstract. *We present AMOS Patches, a large set of image cut-outs, intended primarily for the robustification of trainable local feature descriptors to illumination and appearance changes. Images contributing to AMOS Patches originate from the AMOS dataset of recordings from a large set of outdoor webcams.*

The semiautomatic method used to generate AMOS Patches is described. It includes camera selection, viewpoint clustering and patch selection. For training, we provide both the registered full source images as well as the patches.

A new descriptor, trained on the AMOS Patches and 6Brown datasets, is introduced. It achieves state-of-the-art in matching under illumination changes on standard benchmarks.

1. Introduction

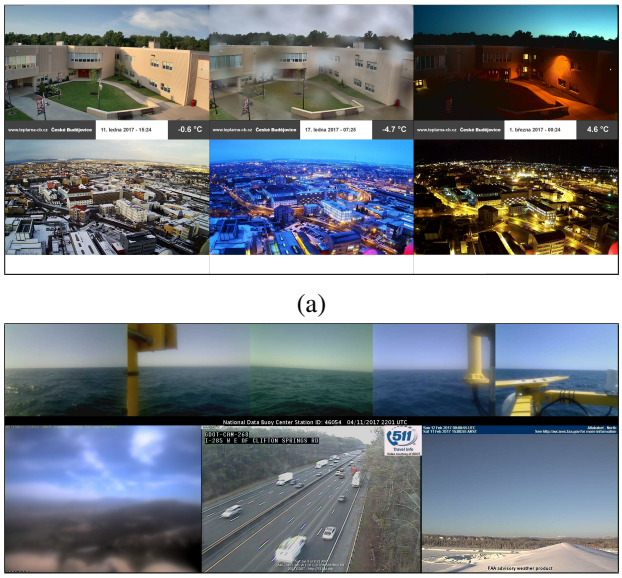
Local feature descriptors are widely used in tasks such as structure from motion [34, 31], image retrieval [36] and in applications like autonomous driving [9], which benefit from the robustness of the descriptors to acquisition conditions.

Recent years have witnessed a noticeable effort to move from handcrafted descriptors [21] to those obtained by deep learning [26, 37]. Existing work explores possible architectures [5, 37], loss functions [26, 13, 16] and improvements of robustness to viewpoint changes by introducing large scale datasets from 3D reconstruction [29, 22].

Robustness to illumination and appearance changes has received little attention, yet it is a bigger challenge for modern descriptors [28, 4]. We tackle this problem by leveraging information from 24/7 webcams located worldwide [15, 14].

We make the following contributions. First, we present a method for extracting veridical patch correspondences from the "static" cameras. Second, we present the AMOS Patches dataset[1] for training of local feature descriptors with improved robustness to changes in illumination and appearance.

As a final contribution, HardNet [26] trained with AMOS Patches achieves state-of-the-art results in the commonly used HPatches benchmark [4].



(a)

(b)

Figure 1: The AMOS dataset [14, 15] - example images from (a) cameras contributing to the AMOS patches set and (b) cameras unsuitable for descriptor training because of blur, dynamic content or dominant sky.

2. Related Work

The literature on local feature descriptors is vast. Here we focus on descriptors which are robust to

[1]The dataset and contributing images are available at https://github.com/pultarmi/AMOS_patches

illumination and appearance changes, refering the reader to Csurka *et al*. [8] for detailed survey on recent advances in local features. There are two main ways towards achieving robustness to illumination change: by descriptor construction and by learning on the appropriate dataset. Normalization of the patch mean and variance is a simple but powerful method, which is implemented in both SIFT [21] and modern learned descriptors [37, 26]. The normalization makes the descriptor invariant to affine changes in pixel intensities in the patch. HalfSIFT [17] treats opposite intensity gradient directions as equal, trading off half of the SIFT dimensionality for being contrast reversal invariant. It works well in medical imaging and infrared-vs-visible matching.

The family of order-based descriptors like LIOP [39] or MROGH [10] operates on the relative order of pixel intensities in the patch instead of on the intensities themselves. Relative order (sorting) is invariant to any monotonically increasing intensity transformation. Descriptors like SymFeat [12], SSIM [35] and learned DASC [18] encode local symmetries and self-similarities. Another possibility is, instead of constructing a descriptor, to apply some transformation to the pixel intensities as done by the learned RGB2NIR [41] or hand-crafted LAT [32], and then use a standard descriptor, e.g. SIFT.

Data-driven approaches mostly include Siamese convolution networks with modality-specific branches, like the Quadruplet Network [3]. The decision which branch to use for a specific patch comes from an external source or a domain classifier. HNet [20] uses an auto-encoder network and style transfer methods like CycleGAN [43] for emulating different modalities.

There is a number of image-level datasets specifically designed for testing illumination-robust recognition: DTU Robot [2], OxfordAffine [25], RobotCar dataset [23], Aachen Day-Night [33], GDB [40], SymBench [12], etc. Despite the importance of the topic, the number of patch-level datasets for illumination-robust descriptors is small, especially those which are suitable for descriptor learning. To our best knowledge, Two Yosemite sequences from the Phototour dataset [6] and the Illumination split of the HPatches dataset [4] are the only ones suitable for descriptor learning and are publicly available.

3. Creating AMOS Patches

AMOS [14, 15] is a continuously growing publicly available dataset collected from outdoor webcams, currently containing over one billion (or 20 TB) images. It is organized into individual camera directories, which are split into folders according to the year and month of the acquisition. The size of the images varies, and so does their quality and the number of images in each camera directory. A typical AMOS camera is static and has approximately 300 times 300 pixel size. Many cameras store images in all seasons and during the whole day.

The advantage of static cameras lies in the fact that they show the same structure under different weather and lighting conditions. Therefore, if observing a static scene, they are highly suitable for training of local feature descriptor robust to illumination and appearance changes.

We learned the hard way that using this type of data is not trivial. Firstly, due to the dataset size, it is not feasible with moderate computing power to load such data into memory. Moreover, preprocessing would take a prohibitive amount of time. Secondly, the training procedure is sensitive to misregistration of the images and the presence and size of moving objects. Many cameras experience technical issues such as: being out of focus, rotating over time, displaying highly dynamic scene (e.g. sky, sea waves), which all significantly hurt the performance of the trained descriptor, as discussed later.

Therefore, we developed a pipeline for the creation of AMOS Patches, shown in Figure 2, which entails several steps to create a clean dataset with veridical patch correspondences. These methods focus on the selection of cameras and images, detection of view switching in a camera and the registration of images. Because of several not easily detectable problems, it was still necessary to perform final manual check of the selected image sets.

3.1. Camera selection

The first step — camera selection — aims at choosing a subset of cameras which are suitable for training, i.e. do not produce very dark images, are sharp and do not display moving objects like cars or boats.

The procedure uses two neural networks, a sky detector [24] and an object detector [1], and computes simple statistics for each of 20 randomly chosen images in each camera directory.

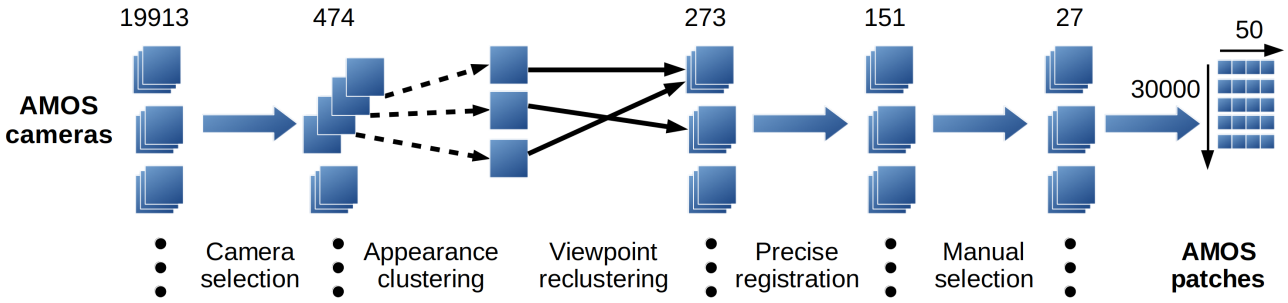


Figure 2: The pipeline of AMOS patches consists of: camera selection to filter out dynamic or empty scenes, appearance clustering to remove redundant images, viewpoint reclustering to tackle switching cameras, precise registration for further filtering, manual pruning for final selection of views and patch sampling.

The camera selection took approximately one week on a single PC (Intel$^{\text{X}}$eon$^{\text{C}}$PU E5-2620) with one GPU GTX Titan X. Processing more images by the neural network detectors increases both the precision of the method and the running time. Our choice is therefore based on the available computation power.

Each image is then checked whether it satisfies the following conditions:

- f_1 : sky area $< 50\%$ *not empty*
- f_2 : no detected cars or boats *not dynamic*
- f_3 : $\text{Var}(\nabla^2 \text{ image}) \geq 180$ *sharp*
- f_4 : mean pixel intensity > 30 *not black*
- f_5 : image size $> (700, 700)$ *large*

A camera is kept if at least 14 out of the 20 images pass the check.

The filter f_5 is the most restrictive, it removes 91% of the cameras – AMOS contains mostly low resolution images. The reasoning behind using f_5 is that images with smaller size often observe a motorway or are blurred. Also, such cameras would not generate many patches. We want to select only a relatively small subset of the cameras with the predefined characteristics and therefore an incorrect removal of a camera is not a problem.

Several cameras were removed because of corrupted image files. The resulting set contains 474 camera folders which were subject to subsequent preprocessing.

3.2. Appearance clustering by K-means

The resulting data is of sufficient quality, but it is highly redundant: images shot in 10 minute intervals are often indistinguishable and very common. To select sufficiently diverse image sets, we run the K-means clustering algorithm with K=120 to keep the most representative images. We use the fc6 layer of the ImageNet-pretrained AlexNet [19] network as the global image descriptor. While not being the state-of-the-art, AlexNet is still the most effective architecture in terms of speed [7], with an acceptable quality.

At this stage of the pipeline, there are K=120 images for each of the C=474 cameras selected, a feasible number for training with the computational resources available.

Feature descriptor training with patches selected from this image set was not successful. We were unable to achieve accuracy higher than 49.1 mean average precision (mAP) in the HPatches matching task; the state-of-the-art is 59.1 mAP – GeoDesc [22].

3.3. Viewpoint clustering with MODS

After examining the data closely, we found that many of the cameras switch between a few views, which breaks our assumption for the generation of ground truth correspondences via identity transformation. In order to filter out the non-matching views, we run MODS [27], a fast method for two-view matching, and split each camera folder into clusters, called views, by applying a threshold on the number of inliers and the difference between the homography matrix and the identity transform.

Let $(x_1, x_2, ..., x_K)$ be a set of images in a camera folder in arbitrary order. MODS matching is first run on pairs $(x_1, x_2), (x_1, x_3), ...(x_1, x_K)$. Image x_1 becomes the reference image in a newly created view, which contains x_i for which the registration yields more than 50 inliers and $\text{SAD}(H(x_1, x_i), I_3) < 50$. SAD denotes the sum of absolute differences, H de-

notes a homography matrix normalized by the element in position $(3, 3)$, I_3 is 3x3 identity matrix. All images in the created view are then removed from the processed image set. The step is repeated until no images remain.

We observed that the number of the resulting views in one camera folder depends on phenomena other than camera movement. For example, in cases where there is a fog or very rainy weather, MODS fails to match most of the image pairs and many of them form a single element cluster, which is excluded from further processing. For each camera, we keep only the view with the largest number of images, if it has more than 50. Each remaining view is reduced to 50 images by random selection.

3.4. Registration with GDB-ICP

While the MODS method is effective in matching and subsequent reclustering of camera sequences, in most of the cases the estimate of the global homography is not suficiently precise. MODS often outputs a homography valid for only small area in the image, see the example shown in Figure 3. Therefore, the views contain also images which are not correctly aligned. To alleviate the problem, we run Generalized Dual Bootstrap-ICP [40] to prune the set of views, keeping those where this second registration is successful.

The registration proceeds as follows. Each view folder contains images $(x_1, x_2, ..., x_{50})$, where image x_1 is the MODS reference. The GDB-ICP registration is run on pairs $(x_1, x_2), (x_1, x_3), ...(x_1, x_{50})$ and warped images $x'_2, x'_3, ..., x'_{50}$ are obtained. If registration fails on any pair, the whole view is removed.

After the precise registration with GDB-ICP, 151 views remained. It is feasible to manually inspect such a set.

3.5. Manual pruning

A few problems remain, see Figure 4, such as dynamic scenes, undetected sky (the sky detector fires mostly on the clear blue sky). As a precaution, we also removed views with very similar content and views from different cameras observing the same place from a different viewpoint. We tried to use the scene segmentation network [42] to detect moving objects, but the result was not satisfactory. The final selection is therefore done by hand, resulting in a set of 27 folders with 50 images each.

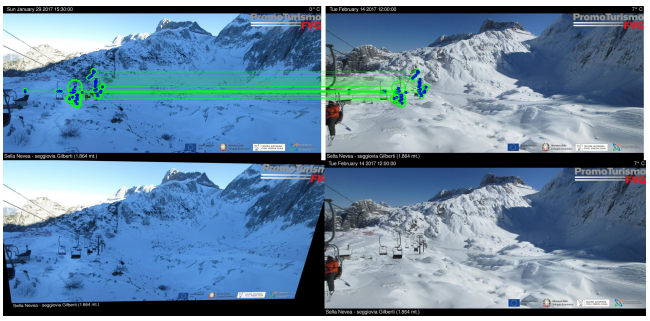


Figure 3: MODS registration failure, most of the correspondences are on moving structures. Top: an image pair with marked inliers. Bottom: wrongly transformed image (left) and the reference.

Figure 4: Manually pruned views. Examples of dynamic scenes (left, center) and a cloud-dominated scene not removed during camera selection (right).

3.6. Patch selection, training

The last phase of the AMOS Patches pipeline consists of sampling images to obtain patch centers, scales and angles, and subsequent cropping from source images. We tested two approaches. First, one may average the images in a view and evaluate a response function over the resulting image. Second, one may evaluate the response function over all images in a view and average the outputs. The resulting 2D map is then used as a probability mask for the selection of patch centers. Scales and angles are sampled independently at random from a predefined range.

For training, we use the hard-in-batch triplet margin loss [26]. This structured loss requires corresponding (positive) pairs of patches on input. Therefore, AMOS Patches dataset consists of sets of patches cropped from the same position in each image in a view. The size of each patch set is equal to the number of images in a view directory, which is 50 in our case. Each patch is resampled to 96 times 96 pixels.

During training, we apply random affine transformation and cropping to get patches of smaller size. First, random rotation from range $(-25°, 25°)$, scaling from range $(0.8, 1.4)$ and shear are applied. Second, from a 64 times 64 center of a patch we crop a

32 times 32 region with random scale. These transformed patches are the input for training.

We use the HardNet implementation in Pytorch [30]. For training we use batch size of 1024, 20 epochs, learning rate = 20, SGD optimizer with momentum $= 0.9$.

4. Evaluating influences on precision

We examine the influence of several choices made before and during training. They relate to batch formation, patch selection and the dataset size. Also, we show the importance of registration of images in a view.

Two evaluation tasks are considered. In the matching task, there are two equally sized sets of patches from two different images. The descriptor is used to find a bijection between them. The average precision (AP) over discrete recall levels is evaluated for each such pair of images. Averaging the results over a number of image pairs gives mAP (mean AP). In the verification task there is a set of pairs of patches. The descriptor assigns a score that the two patches in a pair correspond. Precision-recall curve is then plotted based on the sorted (according to the score) list of patch pairs distances.

4.1. Registration

In this experiment we show the importance of the precise alignment of images. We displace each patch by different shifts and observe the influence on the HPatches matching score, see Figure 5. Notice how the performance of the descriptor improves with a small shift, but then quickly deteriorates. We use $\#$source views $= 27$ (all), 30000 patch sets and Hessian weighting without averaging. These parameters are defined below.

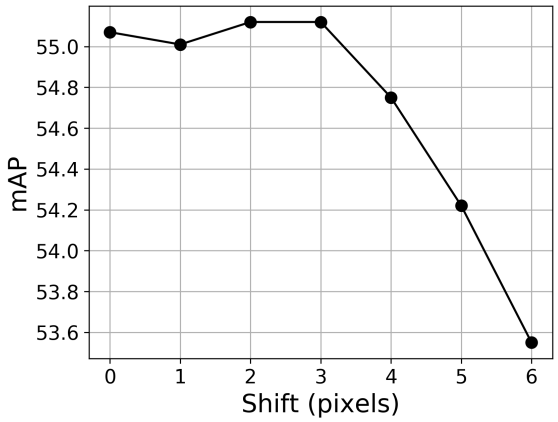


Figure 5: HPatches matching. The mAP score of Hardnet trained on AMOS patches displaced by different shifts.

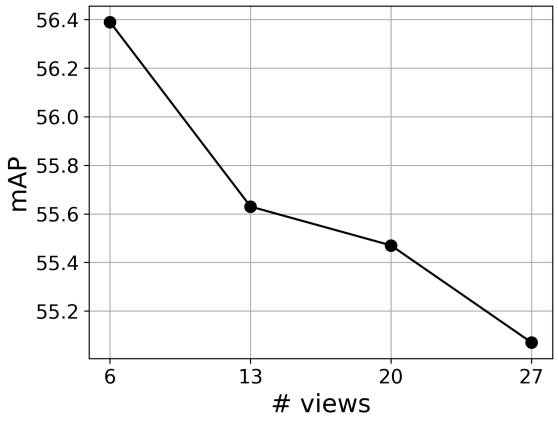


Figure 6: HardNet mAP score in HPatches matching task as a function of the number of source views for a batch. Views are selected randomly in each iteration. Dataset consists of 27 views in total.

4.2. Number of source views

The hard-in-batch triplet margin loss is influenced by the composition of a batch. This experiment shows that lowering the number of views from which we choose patches to form a batch is an effective way to improve training on AMOS Patches, see Figure 6. We interpret this behaviour as follows. Reducing the number of views increases the number of negative patches from the same scene, which are often the most difficult to distinguish.

4.3. AMOS Patches size

Here we examine the influence of the dataset size, i.e. the number of patch sets created from source views, see Figure 7. We use the results from the previous experiment and choose #(source cameras) $= 6$. The graph shows there is a rough increase in HPatches matching score on bigger datasets. Based on the result, we fix the number of patches to be 30 000 to trade off dataset compactness for slightly higher performance.

4.4. Patch sampling

The patch selection method is partially determined by two independent choices: the response function and the averaging method. First, we find the best response function (Table 1), then we keep it fixed

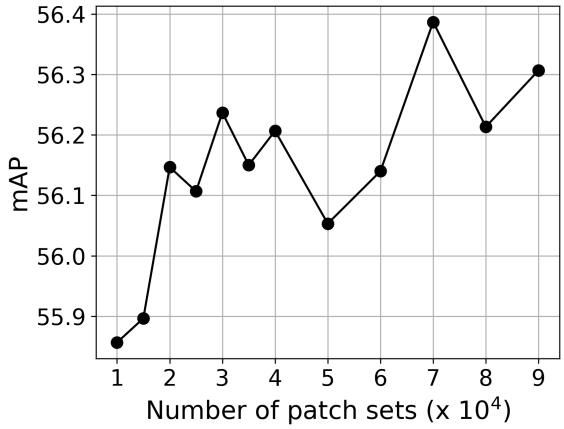


Figure 7: HardNet mAP score in HPatches matching task evaluated for different sizes of AMOS patches training dataset. Each value is an average over 3 different randomly generated datasets of the same size.

and determine the optimal averaging function, which may apply either to outputs from the response function (Table 2) or to images in a view (Table 3).

Table 1: Patch sampling: Influence of the response function on HPatches matching score (mAP).

Weighting	mAP
Uniform	56.20
Hessian	56.39
$\sqrt{\text{Hessian}}$	**56.49**
NMS($\sqrt{\text{Hessian}}$)	56.18

Table 2: Patch sampling: Influence of the response averaging on HPatches matching score (mAP). Weighting function is $\sqrt{\text{Hessian}}$.

Averaging	mAP
none	**56.49**
mean	56.10
median	56.45

5. Evaluation

HPatches and AMOS benchmarks. The evaluation shows that HardNet trained on AMOS Patches and 6Brown dataset outperforms the state-of-the-art descriptors for matching under illumination changes. We also use the new AMOS Patches testing split

Table 3: Patch sampling: Influence of the image averaging on HPatches matching score (mAP). Weighting function is $\sqrt{\text{Hessian}}$.

Image	mAP
random	56.49
median	56.44
mean	**56.58**

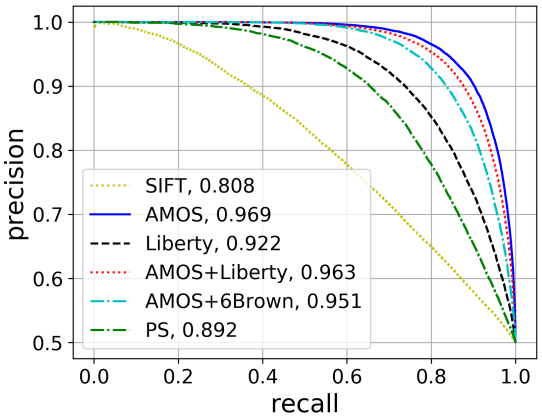


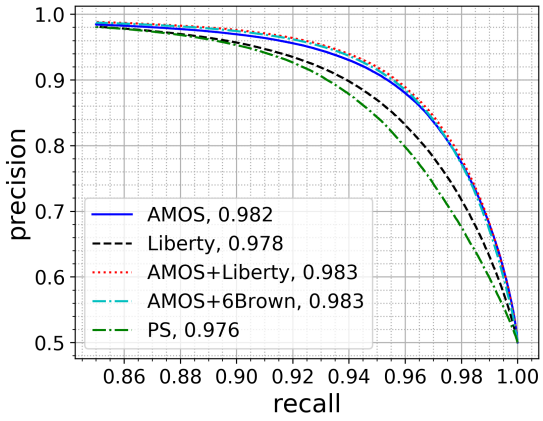
Figure 8: HardNet performance on the AMOS test set, when trained on the AMOS, Liberty, AMOS and Liberty, AMOS and 6Brown and PS [29] datasets. SIFT results are provided as a baseline.

to evaluate robustness to lighting and season-related conditions. See Table 4 for results in the matching task, Figure 9 in the verification task and Figure 8 for comparison on the proposed AMOS Patches test split.

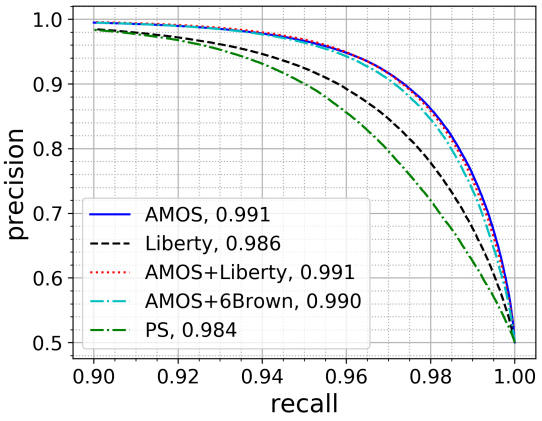
Table 4: HPatches matching scores (mAP).

Training set	HPatches subset		
	illum	view	full
Liberty	49.86	55.62	52.79
6Brown	52.39	59.15	55.83
PS	48.55	**67.43**	58.16
Webcam [38]	51.82	50.77	51.29
AMOS-patches	55.17	57.94	56.58
+Liberty	56.14	60.27	58.24
+6Brown	**56.22**	61.50	**58.91**

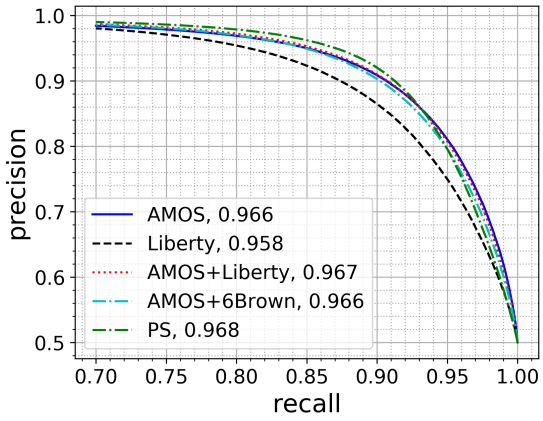
Wide baseline stereo. Finally, we evaluate the descriptors on a real-world task – wide baseline stereo



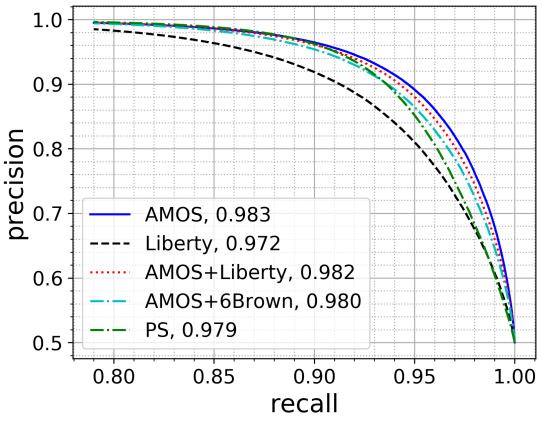
(a) HPatches intra easy



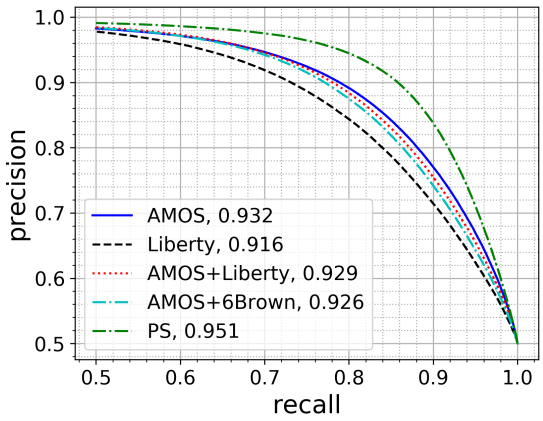
(b) HPatches inter easy



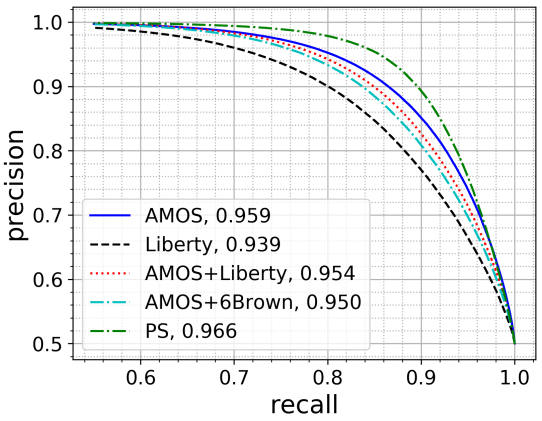
(c) HPatches intra hard



(d) HPatches inter hard



(e) HPatches intra tough



(f) HPatches inter tough

Figure 9: HardNet performance evaluated on the HPatches benchmark. Precision-recall curve is presented based on the output from the verification task. Legend shows the training set name with the corresponding AUC.

Table 5: Comparison of the AMOS+6Br HardNet vs. HardNet++ [26] following the protocol [28]. The number of matched image pairs is shown. The numbers of image pairs in a dataset are boxed. Best results are in **bold**.

Descriptor	EF [44] 33	EVD [27] 15	OxAff [25] 40	SymB [12] 46	GDB [40] 22	map2photo [28] 6	LTLL [11] 172
HardNet++ [26]	31	**15**	**40**	40	18	2	**108**
HardNetAMOS+6Br	**33**	**15**	**40**	**45**	**19**	**4**	106

on multiple datasets, following the protocol [28]. Two metrics are reported: the number of successfully matched image pairs and the average number of inliers per matched pair. Results are shown in Table 5. Edge Foci (EF) [44], Extreme view [27] and Oxford Affine [25] benchmarks provide a sanity check — the performance on the benchmark is saturated and they contain (mostly) images taken from a slightly different viewpoint.

SymB [12], GDB [40] and map2photo [28] contain image pairs which are almost perfectly registered, but have severe differences in illumination or modalities, e.g. drawing vs. photo, etc. AMOS+6Br HardNet performs better than baseline HardNet++ on such datasets. The last dataset – LTLL [11] consists of historical photos and old postcards. The landmarks are depicted with significant changes in both viewpoint and illumination. Baseline HardNet++ slightly outperforms our descriptor. Overall, the benchmark confirms that HardNet trained on AMOS Patches is robust to illumination and appearance changes in real-world scenarios.

6. Conclusion

We provide the AMOS Patches dataset for robustification of local feature descriptors to illumination and appearance changes. It is based on registered images from selected cameras from the AMOS dataset. It has both the training and testing split.

We introduce the local feature descriptor trained on AMOS Patches and 6Brown datasets, which achieves the score of 58.91 mAP in HPatches matching task in full split, compared to the current state-of-the-art: 59.1 mAP (GeoDesc). The advantage of the descriptor is the robustness to illumation. It achieves the state-of-the-art score of 56.22 mAP in matching task, illum split, compared to 52.39 mAP of HardNet++.

We conclude with a list of observations and recommendations related to using webcams for descriptor learning:

- Scene parsing methods do not work well in outdoor webcams. The precision of the near state-of-the-art network [42] is not satisfactory.
- For camera selection we recommend to adopt strict "quality" criteria and be prepared to loose many suitable cameras in the process.
- When picking cameras for training manually, a small and diverse subset is better than a bigger one with similar views or imprecise alignment of images.

Acknowledgements

The authors were supported by the Austrian Ministry for Transport, Innovation and Technology, the Federal Ministry of Science, Research and Economy, and the Province of Upper Austria in the frame of the COMET center SCCH, the CTU student grant SGS17/185/OHK3/3T/13, and the OP VVV funded project CZ.02.1.01/0.0/0.0/16_019/0000765 Research Center for Informatics.

References

[1] https://github.com/kuangliu/torchcv. reviewed on December 2018. 2

[2] H. Aanæs, A. Dahl, and K. Steenstrup Pedersen. Interesting interest points. *International Journal of Computer Vision*, 97:18–35, 2012. 2

[3] C. A. Aguilera, A. D. Sappa, C. Aguilera, and R. Toledo. Cross-spectral local descriptors via quadruplet network. *Sensors*, 17(4), 2017. 2

[4] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk. HPatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 4, page 6, 2017. 1, 2

[5] V. Balntas, E. Riba, D. Ponsa, and K. Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2016. 1

[6] M. Brown and D. G. Lowe. Automatic panoramic image stitching using invariant features. *International Journal on Computer Vision*, 74(1):59–73, 2007. 2

[7] A. Canziani, A. Paszke, and E. Culurciello. An analysis of deep neural network models for practical applications. *arXiv preprint arXiv:1605.07678*, 2016. 3

[8] G. Csurka and M. Humenberger. From handcrafted to deep local invariant features. *arXiv preprint arXiv:1807.10254*, 2018. 2

[9] A. Dewan, T. Caselitz, and W. Burgard. Learning a local feature descriptor for 3d lidar scans. *arXiv preprint arXiv:1809.07494*, 2018. 1

[10] B. Fan, F. Wu, and Z. Hu. Aggregating gradient distributions into intensity orders: A novel local image descriptor. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2011. 2

[11] B. Fernando, T. Tommasi, and T. Tuytelaars. Location recognition over large time lags. *Computer Vision and Image Understanding*, 139:21 – 28, 2015. 8

[12] D. Hauagge and N. Snavely. Image matching using local symmetry features. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 206–213, 2012. 2, 8

[13] K. He, Y. Lu, and S. Sclaroff. Local descriptors optimized for average precision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1

[14] N. Jacobs, W. Burgin, N. Fridrich, A. Abrams, K. Miskell, B. H. Braswell, A. D. Richardson, and R. Pless. The global network of outdoor webcams: properties and applications. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 111–120. ACM, 2009. 1, 2

[15] N. Jacobs, N. Roman, and R. Pless. Consistent temporal variations in many outdoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–6. IEEE, 2007. 1, 2

[16] M. Keller, Z. Chen, F. Maffra, P. Schmuck, and M. Chli. Learning deep descriptors with scale-aware triplet networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1

[17] A. Kelman, M. Sofka, and C. V. Stewart. Keypoint descriptors for matching across multiple image modalities and non-linear intensity variations. In *CVPR 2007*, 2007. 2

[18] S. Kim, D. Min, B. Ham, M. N. Do, and K. Sohn. Dasc: Robust dense descriptor for multi-modal and multi-spectral correspondence estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9), Sept 2017. 2

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 3

[20] W. Liu, X. Shen, C. Wang, Z. Zhang, C. Wen, and J. Li. H-net: Neural network for cross-domain image patch matching. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, 7 2018. 2

[21] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 1, 2

[22] Z. Luo, T. Shen, L. Zhou, S. Zhu, R. Zhang, Y. Yao, T. Fang, and L. Quan. Geodesc: Learning local descriptors by integrating geometry constraints. In *The European Conference on Computer Vision (ECCV)*, September 2018. 1, 3

[23] W. Maddern, G. Pascoe, C. Linegar, and P. Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*, 36(1):3–15, 2017. 2

[24] R. P. Mihail, S. Workman, Z. Bessinger, and N. Jacobs. Sky segmentation in the wild: An empirical study. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–6. IEEE, 2016. 2

[25] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision (IJCV)*, 65(1):43–72, 2005. 2, 8

[26] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *Advances in Neural Information Processing Systems*, pages 4826–4837, 2017. 1, 2, 4, 8

[27] D. Mishkin, J. Matas, and M. Perdoch. Mods: Fast and robust method for two-view matching. *Computer Vision and Image Understanding*, 141:81–93, 2015. 3, 8

[28] D. Mishkin, J. Matas, M. Perdoch, and K. Lenc. Wxbs: Wide baseline stereo generalizations. In *Proceedings of the British Machine Vision Conference (BMVC)*, September 2015. 1, 8

[29] R. Mitra, N. Doiphode, U. Gautam, S. Narayan, S. Ahmed, S. Chandran, and A. Jain. A large dataset for improving patch matching. *arXiv preprint arXiv:1801.01466*, 2018. 1, 6

[30] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017. 5

[31] A. Resindra, A. Torii, and M. Okutomi. Structure from motion using dense cnn features with keypoint relocalization. *IPSJ Transactions on Computer Vision and Applications*, 10, Dec 2018. 1

[32] S. Ryu, S. Kim, and K. Sohn. Lat: Local area transform for cross modal correspondence matching. *Pattern Recognition*, 63, 2017. 2

[33] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, F. Kahl, and T. Pajdla. Benchmarking 6DOF outdoor visual localization in changing conditions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8601–8610. IEEE Computer Society, 2018. 2

[34] J. L. Schonberger and J.-M. Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016. 1

[35] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007. 2

[36] T. Shen, Z. Luo, L. Zhou, R. Zhang, S. Zhu, T. Fang, and L. Quan. Matchable image retrieval by learning from surface reconstruction. *arXiv preprint arXiv:1811.10343*, 2018. 1

[37] Y. Tian, B. Fan, and F. Wu. L2-net: Deep learning of discriminative patch descriptor in euclidean space. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2

[38] Y. Verdie, K. M. Yi, P. Fua, and V. Lepetit. TILDE: A temporally invariant learned detector. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 6

[39] Z. Wang, B. Fan, G. Wang, and F. Wu. Exploring local and overall ordinal information for robust feature description. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(11), Nov 2016. 2

[40] G. Yang, C. V. Stewart, M. Sofka, and C.-L. Tsai. Registration of challenging image pairs: Initialization, estimation, and decision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(11), 2007. 2, 4, 8

[41] T. Zhi, B. R. Pires, M. Hebert, and S. G. Narasimhan. Deep material-aware cross-spectral stereo matching. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[42] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba. Semantic understanding of scenes through the ADE20k dataset. *International Journal on Computer Vision*, 2018. 4, 8

[43] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 2

[44] C. L. Zitnick and K. Ramnath. Edge foci interest points. In *IEEE International Conference on Computer Vision (ICCV)*, pages 359–366, 2011. 8