2D Feature Detectors and Descriptors

CV&ML reading group, 5/29

Presenter: You-Yi Jau, M.S. in UCSD

Outlines

- Introduction
  - Motivation
  - Recap: Sift, Superpoint
- LF-Net: Learning Local Features from Images
  - Both detector and descriptor
- D2-Net: A Trainable CNN for Joint Detection and Description of Local Features
  - Both detector and descriptor
- ContextDesc: Local Descriptor Augmentation with Cross-Modality Context
  - Descriptor

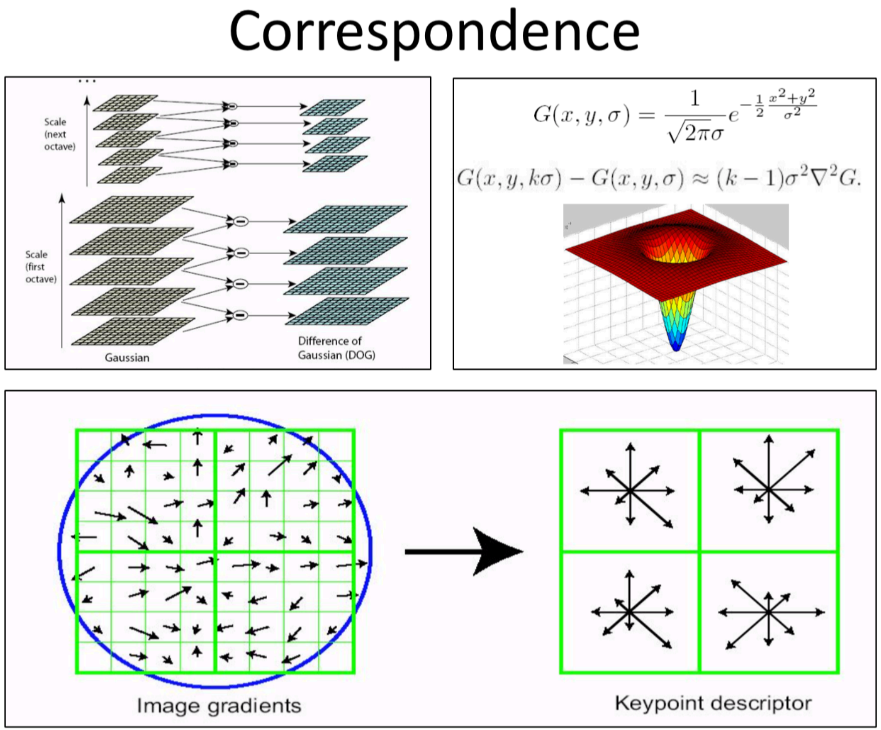
Motivation and problem description

- Why 2D features?
  - Establishing pixel-level correspondences is important
  - Applications in 3D computer vision, video compression, tracking, image retrieval, and visual localization
- Why sparse feature?
  - Correspondences can be matched efficiently via nearest neighbor search

Motivation and problem description

- Why 2D features?
  - Establishing pixel-level correspondences is important
  - Applications in 3D computer vision, video compression, tracking, image retrieval, and visual localization
- Why sparse feature?
  - Correspondences can be matched efficiently via nearest neighbor search
  - Sparse local features have been applied successfully under a wide range of imaging conditions. However, they typically perform poorly under extreme appearance changes.
  - local descriptors can still be matched successfully even if keypoints cannot be detected reliably
  - we propose a describe-and-detect approach to sparse local feature detection and description: Rather than performing feature detection early on based on low-level information, we propose to postpone the detection stage.

SIFT: strong baseline



Correspondence

$$G(x,y,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\frac{x^2+y^2}{\sigma^2}}$$

$$G(x,y,k\sigma) - G(x,y,\sigma) \approx (k-1)\sigma^2\nabla^2 G.$$

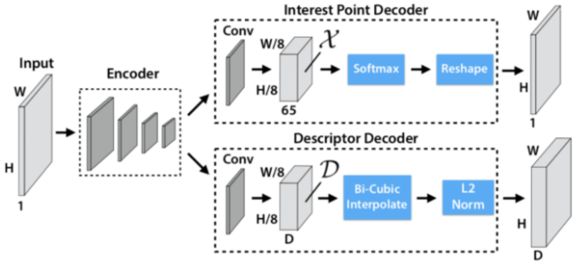
Distinctive image features from scale-invariant keypoints

DG Lowe - International journal of computer vision, 2004

Slide from Manmohan Chandraker.

CSE 252C: Advanced Computer Vision

Superpoint (CVPR’18)



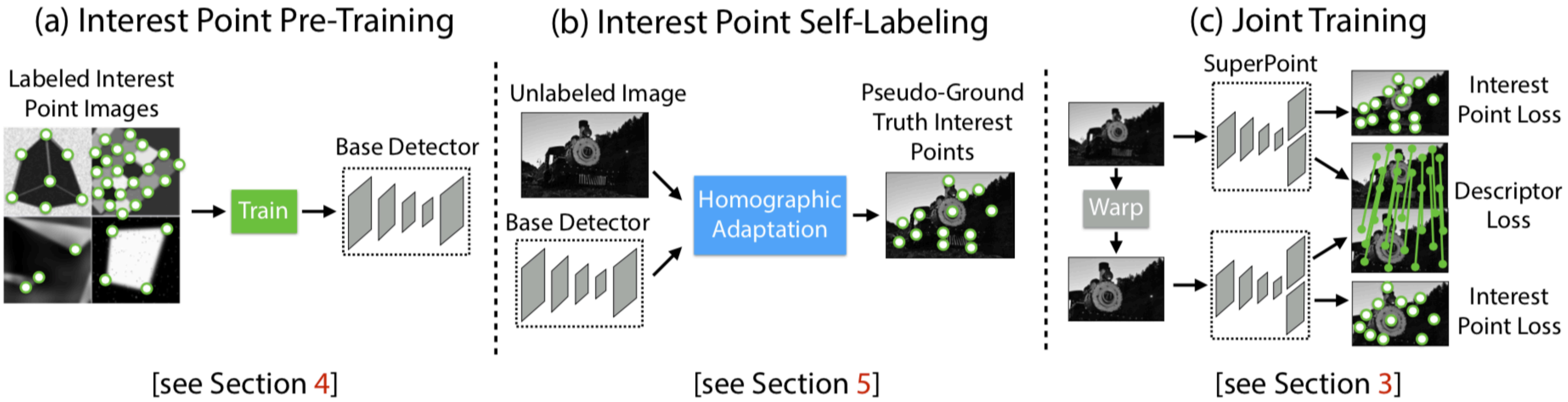
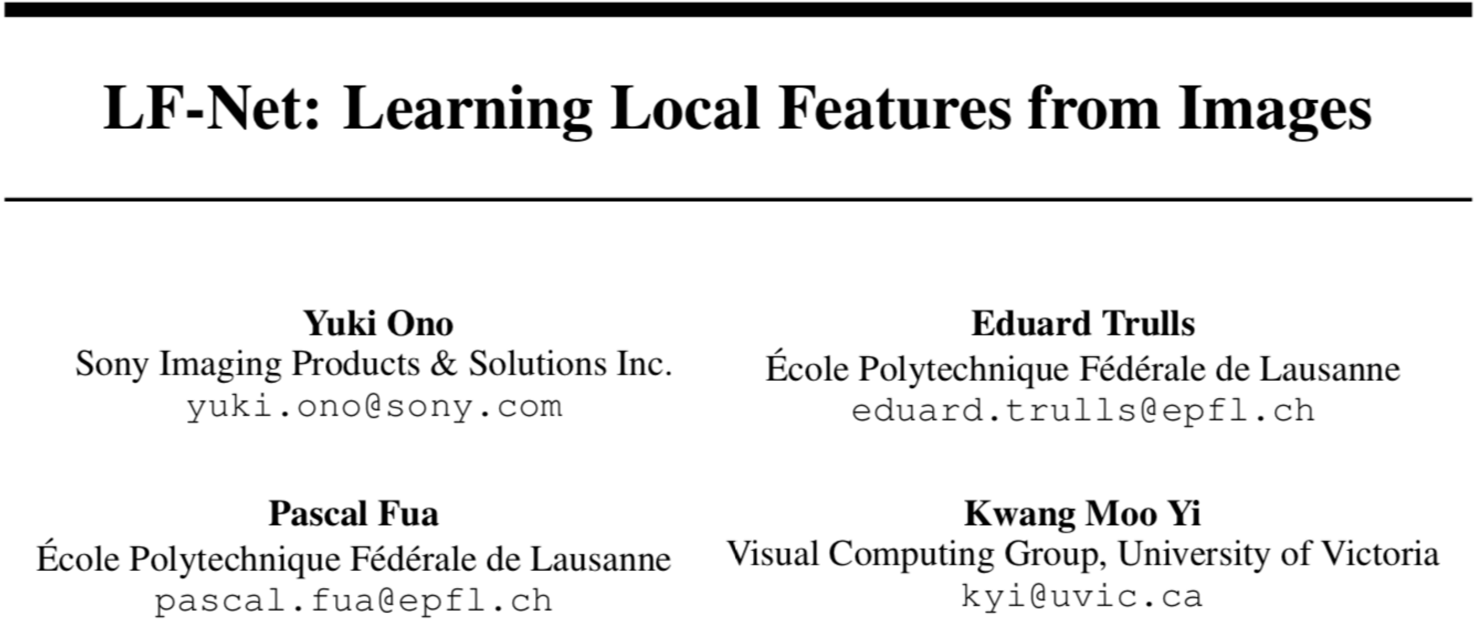


Figure 2. **Self-Supervised Training Overview.** In our self-supervised approach, we (a) pre-train an initial interest point detector on synthetic data and (b) apply a novel Homographic Adaptation procedure to automatically label images from a target, unlabeled domain. The generated labels are used to (c) train a fully-convolutional network that jointly extracts interest points and descriptors from an image.



LF-Net: Learning Local Features from Images

Yuki Ono
Sony Imaging Products & Solutions Inc.
yuki.ono@sony.com

Eduard Trulls
École Polytechnique Fédérale de Lausanne
eduard.trulls@epfl.ch

Pascal Fua
École Polytechnique Fédérale de Lausanne
pascal.fua@epfl.ch

Kwang Moo Yi
Visual Computing Group, University of Victoria
kyi@uvic.ca

NIPS 2018

LF-Net: learning local features from **images**
Y Ono, E Trulls, P Fua, KM Yi - Advances in Neural Information Processing Systems, 2018

Highlights

- LF-Net: Local Feature Network
- Contribution
  - Trainable end-to-end
  - use image pairs: relative pose and corresponding depth maps

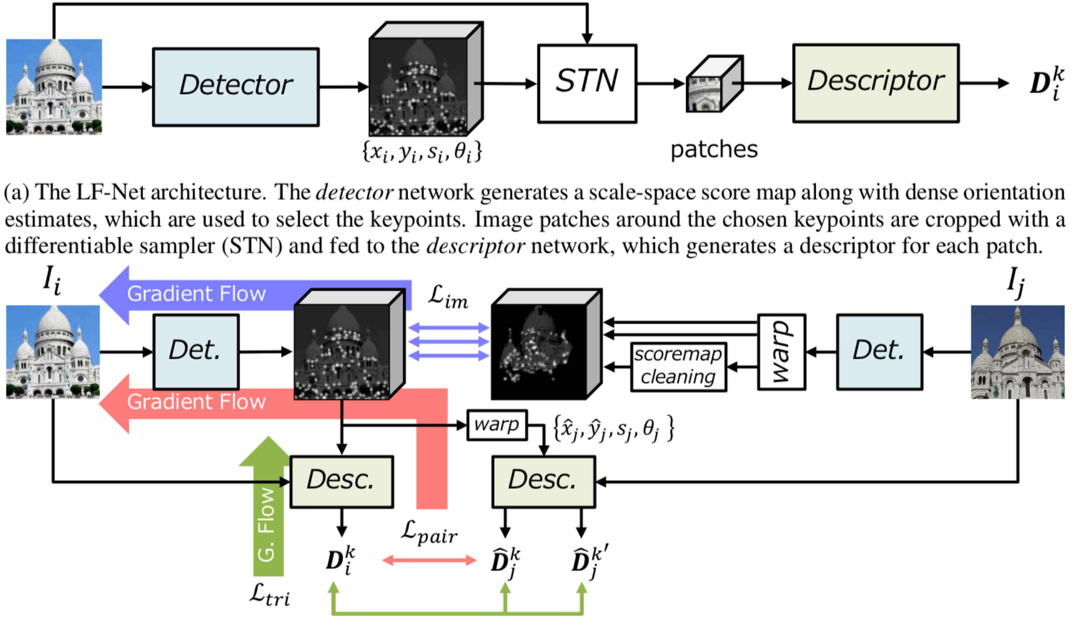
Motivation and problem description

- Contribution
  - we propose a sparse-matching method with a novel deep architecture, which we name LF-Net, for Local Feature Network
  - we use image pairs for which we know the relative pose and corresponding depth maps

Method overview

- learn a local feature pipeline from scratch, using collections of images without the need for human supervision.
- we exploit depth and relative camera pose cues to create a virtual target that the network should achieve on one image

Pipeline

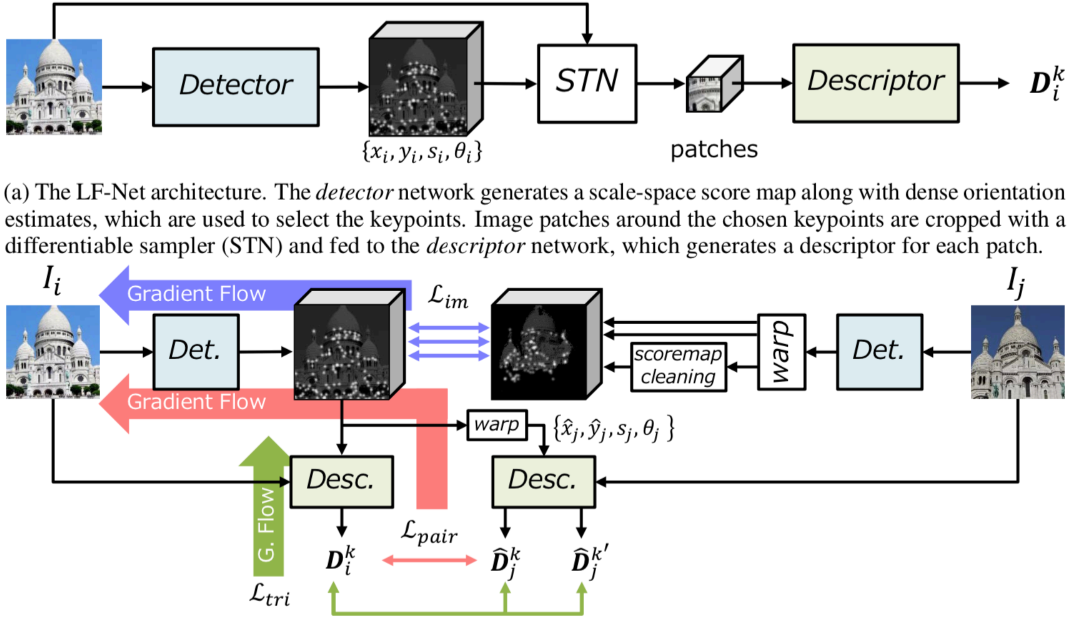


(a) The LF-Net architecture. The *detector* network generates a scale-space score map along with dense orientation estimates, which are used to select the keypoints. Image patches around the chosen keypoints are cropped with a differentiable sampler (STN) and fed to the *descriptor* network, which generates a descriptor for each patch.

(b) For training we use a *two-branch* LF-Net, containing two identical copies of the network, processing two corresponding images I_i and I_j. Branch j (right) is used to generate a supervision signal for branch i (left), created by warping the results from i to j. As this is not differentiable, we optimize only over branch i, and update the network copy for branch j in the next iteration. We omit the samplers in this figure, for simplicity.

Figure 1: (a) The Local Feature Network (LF-Net). (b) Training with two LF-Nets.

Pipeline



(a) The LF-Net architecture. The *detector* network generates a scale-space score map along with dense orientation estimates, which are used to select the keypoints. Image patches around the chosen keypoints are cropped with a differentiable sampler (STN) and fed to the *descriptor* network, which generates a descriptor for each patch.

(b) For training we use a *two-branch* LF-Net, containing two identical copies of the network, processing two corresponding images I_i and I_j. Branch j (right) is used to generate a supervision signal for branch i (left), created by warping the results from i to j. As this is not differentiable, we optimize only over branch i, and update the network copy for branch j in the next iteration. We omit the samplers in this figure, for simplicity.

Figure 1: (a) The Local Feature Network (LF-Net). (b) Training with two LF-Nets.

- Detector loss: $\mathcal{L}_{det} = \mathcal{L}_{im} + \lambda_{pair}\mathcal{L}_{pair} + \mathcal{L}_{geom}$
- Descriptor loss: $\mathcal{L}_{desc} = \mathcal{L}_{tri}$

$$\mathcal{L}_{im}(\mathbf{S}_i, \mathbf{S}_j) = |\mathbf{S}_i - g(w(\mathbf{S}_j))|^2 \ . \qquad (2)$$

$$\mathcal{L}_{pair}(\mathbf{D}_i^k, \hat{\mathbf{D}}_j^k) = \sum_k |\mathbf{D}_i^k - \hat{\mathbf{D}}_j^k|^2 \ .$$

$$\mathcal{L}_{geom}(s_i^k, \theta_i^k, \hat{s}_i^k, \hat{\theta}_j^k) = \lambda_{ori} \sum_k |\theta_i^k - \hat{\theta}_j^k|^2 + \lambda_{scale} \sum_k |s_i^k - \hat{s}_j^k|^2$$

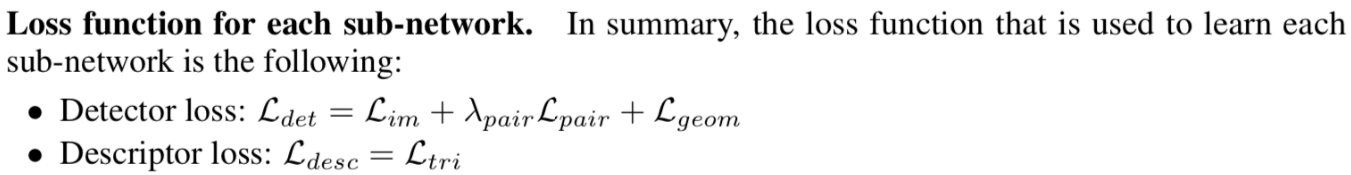
With the matching and non-matching pairs, we form the triplet loss as:

$$\mathcal{L}_{tri}(\mathbf{D}_i^k, \hat{\mathbf{D}}_j^k, \hat{\mathbf{D}}_j^{k'}) = \sum_k \max\left(0, |\mathbf{D}_i^k - \hat{\mathbf{D}}_j^k|^2 - |\mathbf{D}_i^k - \hat{\mathbf{D}}_j^{k'}|^2 + C\right) \ .$$

where $k' \neq k$, *i.e.*, it can be any non-corresponding sample, and C=1 is the margin.

Loss

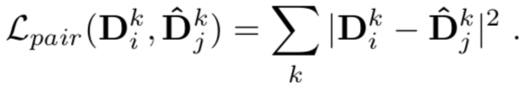
- Final Loss



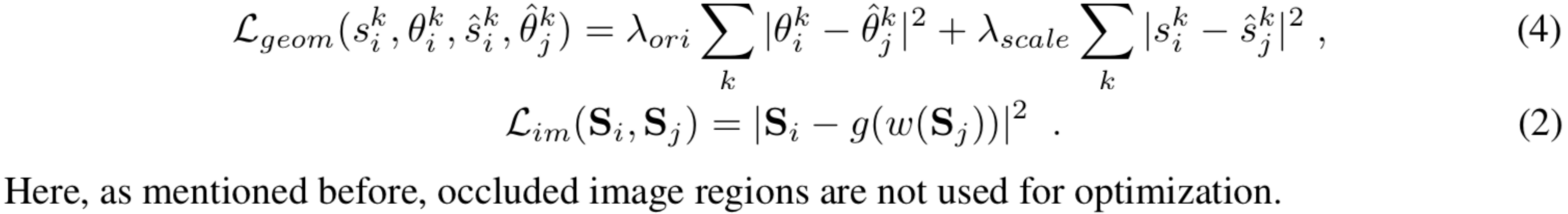
Loss function for each sub-network. In summary, the loss function that is used to learn each sub-network is the following:

- Detector loss: $\mathcal{L}_{det} = \mathcal{L}_{im} + \lambda_{pair}\mathcal{L}_{pair} + \mathcal{L}_{geom}$
- Descriptor loss: $\mathcal{L}_{desc} = \mathcal{L}_{tri}$

- Detector consistency loss: known correspondences



$$\mathcal{L}_{pair}(\mathbf{D}_i^k, \hat{\mathbf{D}}_j^k) = \sum_k |\mathbf{D}_i^k - \hat{\mathbf{D}}_j^k|^2 \ .$$



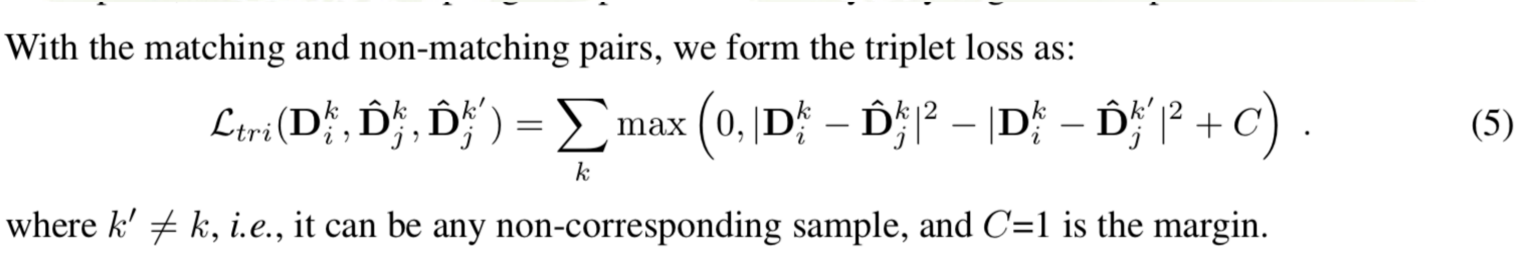
$$\mathcal{L}_{geom}(s_i^k, \theta_i^k, \hat{s}_i^k, \hat{\theta}_j^k) = \lambda_{ori} \sum_k |\theta_i^k - \hat{\theta}_j^k|^2 + \lambda_{scale} \sum_k |s_i^k - \hat{s}_j^k|^2 \ , \tag{4}$$

$$\mathcal{L}_{im}(\mathbf{S}_i, \mathbf{S}_j) = |\mathbf{S}_i - g(w(\mathbf{S}_j))|^2 \quad . \tag{2}$$

Here, as mentioned before, occluded image regions are not used for optimization.

Loss

- Descriptor loss
  - triplet loss



With the matching and non-matching pairs, we form the triplet loss as:

$$\mathcal{L}_{tri}(\mathbf{D}_i^k, \hat{\mathbf{D}}_j^k, \hat{\mathbf{D}}_j^{k'}) = \sum_k \max\left(0, |\mathbf{D}_i^k - \hat{\mathbf{D}}_j^k|^2 - |\mathbf{D}_i^k - \hat{\mathbf{D}}_j^{k'}|^2 + C\right) \ . \tag{5}$$

where $k' \neq k$, *i.e.*, it can be any non-corresponding sample, and C=1 is the margin.

Method details and analysis

- Detector
  - dense, multi-scale, fully convolutional network
  - predict keypoint locations, scales, and orientations
- Descriptor
  - given patches cropped around the keypoints
  - outputs local descriptors

Method details and analysis

- Detector
  - The first network is a dense, multi-scale, fully convolutional network that returns keypoint locations, scales, and orientations.
- Descriptor
  - The second is a network that outputs local descriptors given patches cropped around the keypoints produced by the first network.

Detector details

- Feature map: model
  - ResNet blocks
  - same output size as the input
- Scale
  - 5 layers
  - heatmap → softmax over patches → softmax over different scales → top K pixels as keypoints → softargmax (sub-pixel)
- Orientation estimation
  - 5 x 5 convolution on heatmap → predict cosθ, sinθ

Detector details

- Feature map: model
  - ResNet layout with three blocks
  - Each block contains 5 x 5 convolutional filters followed by batch normalization, leaky-ReLU activations, and another set of 5 x 5 convolutions.
  - same output size as the input, and have 16 output channels
- Scale
  - uniform intervals between 1/R and R, where N = 5 and R = 2^(0.5)
  - heatmap → softmax over patches → softmax over different scales → top K pixels as keypoints → softargmax (sub-pixel)
- Orientation estimation
  - 5 x 5 convolution on heatmap → predict cosθ, sinθ

Descriptor details

- patch size: 32 x 32
- Model:
  - three 3 X 3 convolutional filters with a stride of 2 and 64, 128, and 256 channels respectively.
  - And fully-connected 512-channel layer

Descriptor details

- crop them from the normalized images and resize them to 32 x 32
- Model:
  - Our descriptor network comprises three 3 X 3 convolutional filters with a stride of 2 and 64, 128, and 256 channels respectively. And fully-connected 512-channel layer

Training data

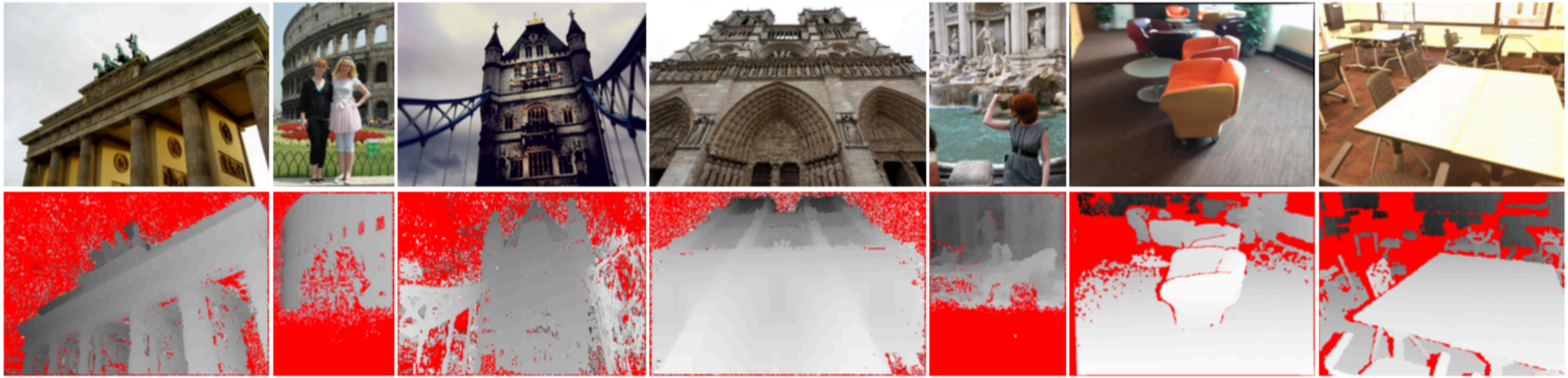
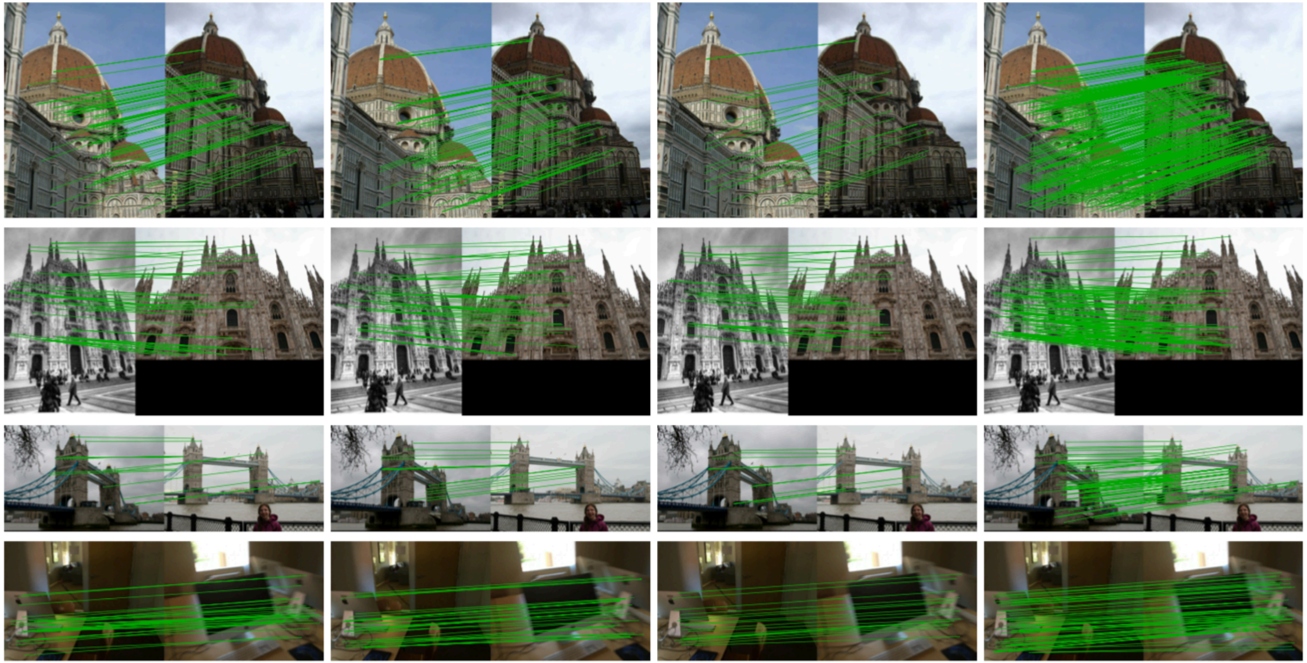


Figure 2: Samples from our indoors and outdoors datasets. Image regions without depth measurements, due to occlusions or sensor shortcomings, are drawn in red, and are simply excluded from the optimization. Note the remaining artefacts in the depth maps for outdoors images.

Qualitative results



(a) SIFT (b) SURF (c) A-KAZE (d) LF-Net (ours)

Figure 3: Qualitative matching results, with correct matches drawn in green.

Matching score on indoor and outdoor

- w/ rot-scl:
  - data augmentation

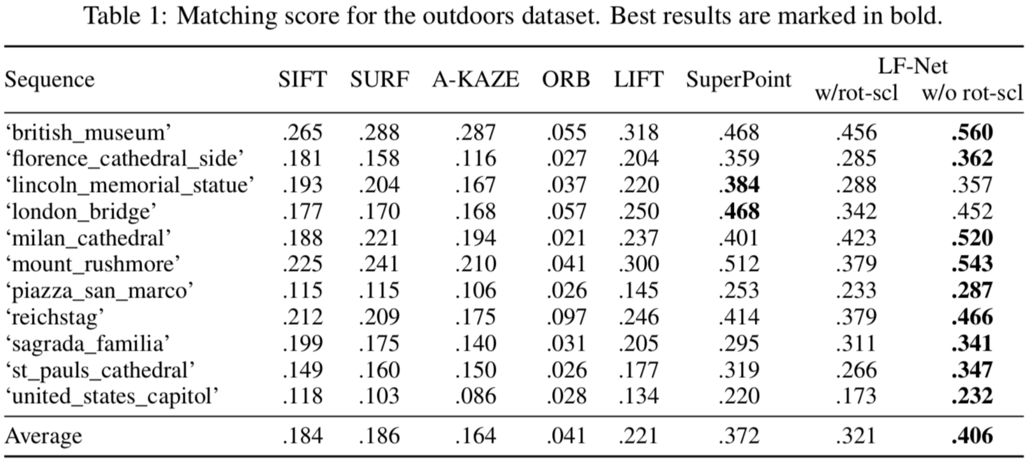


Table 1: Matching score for the outdoors dataset. Best results are marked in bold.

Sequence	SIFT	SURF	A-KAZE	ORB	LIFT	SuperPoint	LF-Net w/rot-scl	LF-Net w/o rot-scl
‘british_museum’	.265	.288	.287	.055	.318	.468	.456	**.560**
‘florence_cathedral_side’	.181	.158	.116	.027	.204	.359	.285	**.362**
‘lincoln_memorial_statue’	.193	.204	.167	.037	.220	**.384**	.288	.357
‘london_bridge’	.177	.170	.168	.057	.250	**.468**	.342	.452
‘milan_cathedral’	.188	.221	.194	.021	.237	.401	.423	**.520**
‘mount_rushmore’	.225	.241	.210	.041	.300	.512	.379	**.543**
‘piazza_san_marco’	.115	.115	.106	.026	.145	.253	.233	**.287**
‘reichstag’	.212	.209	.175	.097	.246	.414	.379	**.466**
‘sagrada_familia’	.199	.175	.140	.031	.205	.295	.311	**.341**
‘st_pauls_cathedral’	.149	.160	.150	.026	.177	.319	.266	**.347**
‘united_states_capitol’	.118	.103	.086	.028	.134	.220	.173	**.232**
Average	.184	.186	.164	.041	.221	.372	.321	**.406**

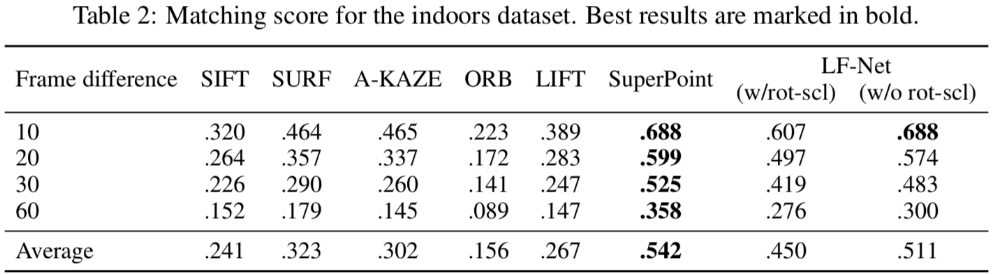
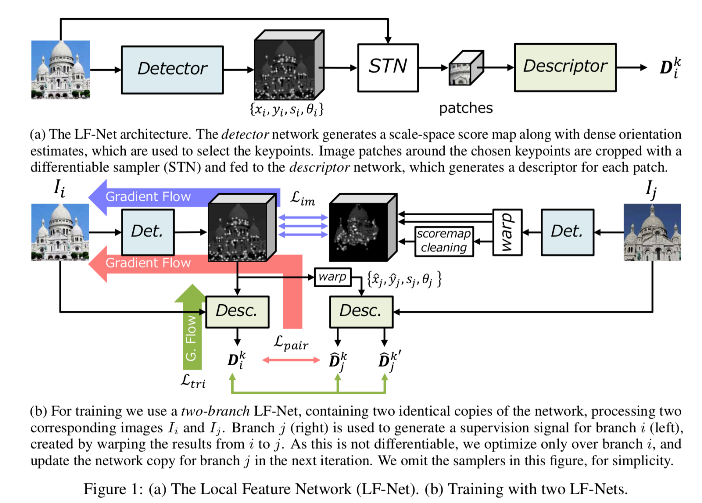


Table 2: Matching score for the indoors dataset. Best results are marked in bold.

Frame difference	SIFT	SURF	A-KAZE	ORB	LIFT	SuperPoint	LF-Net (w/rot-scl)	LF-Net (w/o rot-scl)
10	.320	.464	.465	.223	.389	**.688**	.607	**.688**
20	.264	.357	.337	.172	.283	**.599**	.497	.574
30	.226	.290	.260	.141	.247	**.525**	.419	.483
60	.152	.179	.145	.089	.147	**.358**	.276	.300
Average	.241	.323	.302	.156	.267	**.542**	.450	.511

In short

- LF-Net is deep learning approach for detector and descriptor
- Detector and descriptor do not share parameters
- Use output from off-the-shelf SFM method as supervision



(a) The LF-Net architecture. The *detector* network generates a scale-space score map along with dense orientation estimates, which are used to select the keypoints. Image patches around the chosen keypoints are cropped with a differentiable sampler (STN) and fed to the *descriptor* network, which generates a descriptor for each patch.

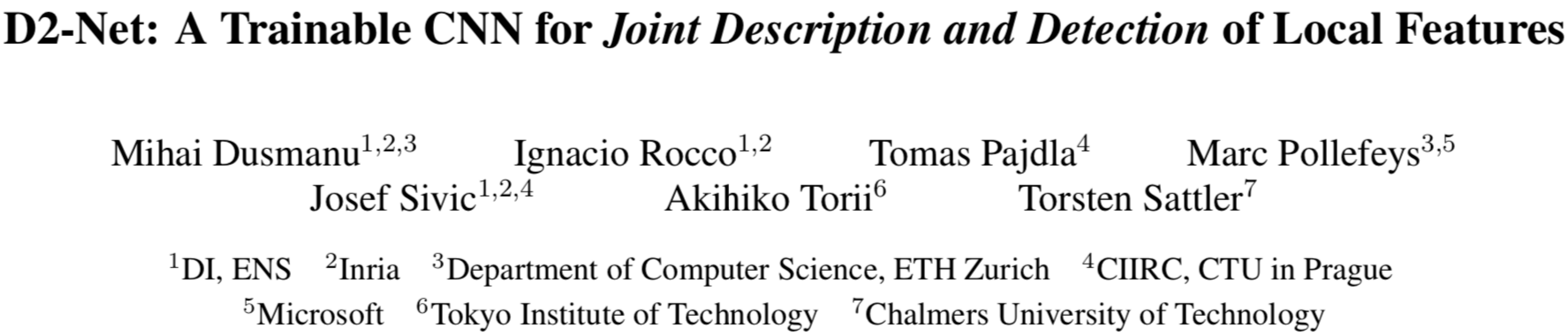
(b) For training we use a *two-branch* LF-Net, containing two identical copies of the network, processing two corresponding images I_i and I_j. Branch j (right) is used to generate a supervision signal for branch i (left), created by warping the results from i to j. As this is not differentiable, we optimize only over branch i, and update the network copy for branch j in the next iteration. We omit the samplers in this figure, for simplicity.

Figure 1: (a) The Local Feature Network (LF-Net). (b) Training with two LF-Nets.

Future work and discussion

- How good can LF-Net work on structure from motion task, compared to SIFT?

Questions?



D2-Net: A Trainable CNN for *Joint Description and Detection* of Local Features

Mihai Dusmanu[1,2,3] Ignacio Rocco[1,2] Tomas Pajdla[4] Marc Pollefeys[3,5]
Josef Sivic[1,2,4] Akihiko Torii[6] Torsten Sattler[7]

[1]DI, ENS [2]Inria [3]Department of Computer Science, ETH Zurich [4]CIIRC, CTU in Prague
[5]Microsoft [6]Tokyo Institute of Technology [7]Chalmers University of Technology

CVPR 2019

D2-Net: A Trainable CNN for Joint Detection and Description of Local Features

M Dusmanu, I Rocco, T Pajdla, M Pollefeys, J Sivic… - arXiv preprint arXiv …, 2019

Motivation and problem description

- Why feature?
  - Establishing pixel-level correspondences is important
  - Applications in 3D computer vision, video compression, tracking, image retrieval, and visual localization
- Why sparse feature?
  - Correspondences can be matched efficiently via nearest neighbor search
  - Sparse local features have been applied successfully under a wide range of imaging conditions. However, they typically perform poorly under extreme appearance changes.
  - local descriptors can still be matched successfully even if keypoints cannot be detected reliably
  - we propose a describe-and-detect approach to sparse local feature detection and description: Rather than performing feature detection early on based on low-level information, we propose to postpone the detection stage.

Qualitative results

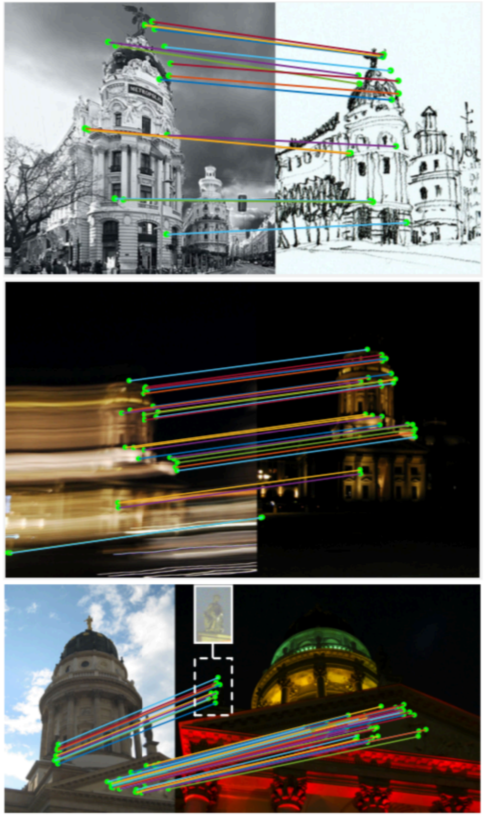


Figure 1: **Examples of matches obtained by the D2-Net method.** The proposed method can find image correspondences even under significant appearance differences caused by strong changes in illumination such as day-to-night, changes in depiction style or under image degradation caused by motion blur.

Main idea

- Change from detect-then-describe to detect-and-describe
- shares all parameters between detection and description
- perform dense feature extraction for both detector and descriptor

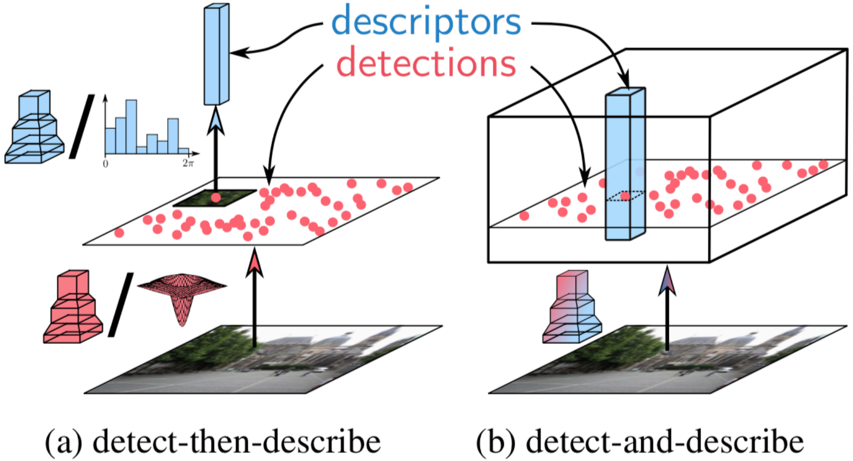


Figure 2: **Comparison between different approaches for feature detection and description.** Pipeline (a) corresponds to different variants of the two-stage detect-then-describe approach. In contrast, our proposed pipeline (b) uses a single CNN which extracts dense features that serve as both descriptors and detectors.

Prior work

- Local features
  - SIFT
  - SuperPoint: shares a deep representation between detection and description
- Dense descriptor extraction and matching.
  - Christopher B. Choy, JunYoung Gwak, Silvio Savarese, andManmohan Chandraker. Universal Correspondence Net-work. In NIPS, 2016.
- Image retrieval.
- Object detection.

Method overview

- shares all parameters between detection and description
- perform dense feature extraction for both detector and descriptor

Method overview

- On the contrary, our method shares all parameters between detection and description and uses a joint formulation that simultaneously optimizes for both tasks.
- Contrary to the classical detect-then-describe ap- proaches, which use a two-stage pipeline, we propose to perform dense feature extraction to obtain a representation that is simultaneously a detector and a descriptor. Because both detector and descriptor share the underlying represen- tation, we refer to our approach as D2.

Pipeline

- Pretrained VGG16 feature extractor as initialization

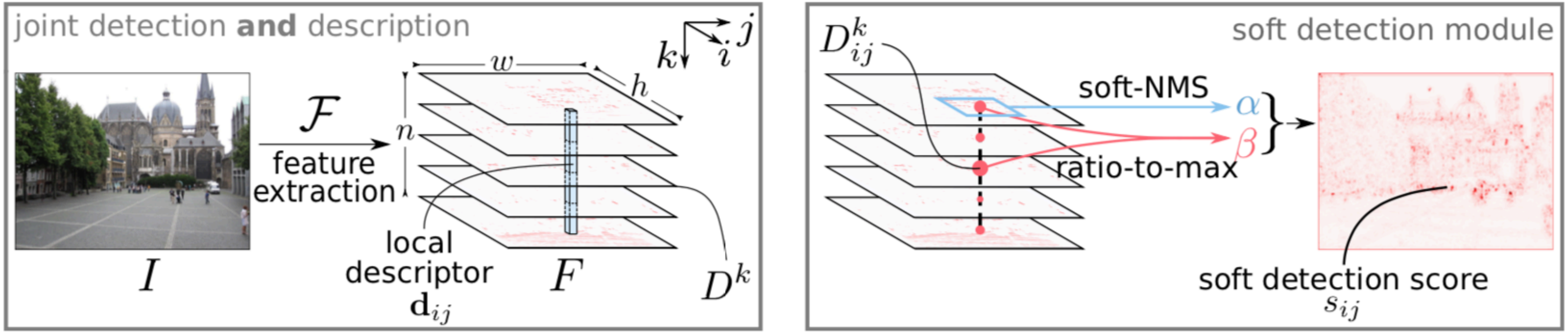


Figure 3: **Proposed detect-and-describe (D2) network.** A feature extraction CNN $\mathcal{F}$ is used to extract feature maps that play a dual role: (i) local descriptors $\mathbf{d}_{ij}$ are simply obtained by traversing all the n feature maps D^k at a spatial position (i, j); (ii) detections are obtained by performing a non-local-maximum suppression on a feature map followed by a non-maximum suppression across each descriptor - during training, keypoint detection scores s_{ij} are computed from a soft local-maximum score α and a ratio-to-maximum score per descriptor β.

Pipeline

- Descriptor
  - directly get from feature map

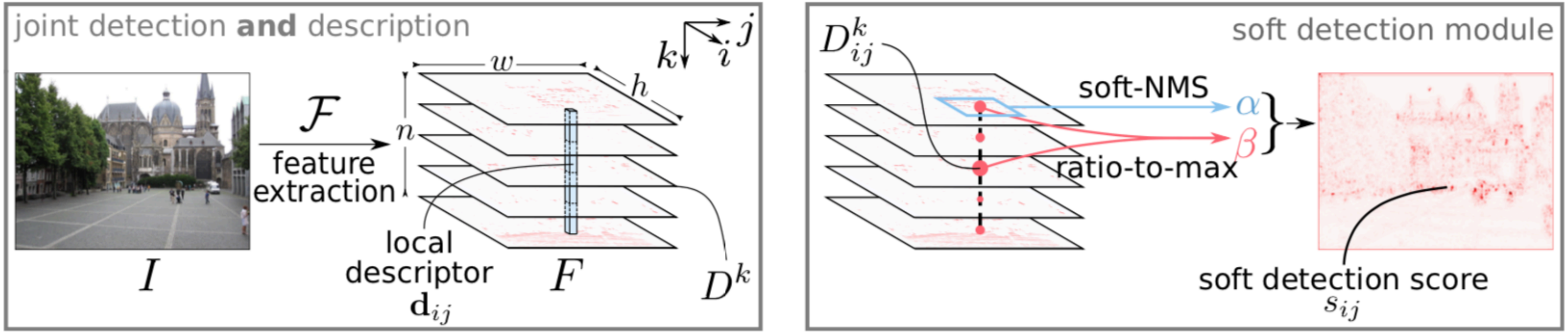


Figure 3: **Proposed detect-and-describe (D2) network.** A feature extraction CNN $\mathcal{F}$ is used to extract feature maps that play a dual role: (i) local descriptors $\mathbf{d}_{ij}$ are simply obtained by traversing all the n feature maps D^k at a spatial position (i, j); (ii) detections are obtained by performing a non-local-maximum suppression on a feature map followed by a non-maximum suppression across each descriptor - during training, keypoint detection scores s_{ij} are computed from a soft local-maximum score α and a ratio-to-maximum score per descriptor β.

Pipeline

- Detector
  - Soft local-max with neighbors → channel-wise NMS → max across feature maps

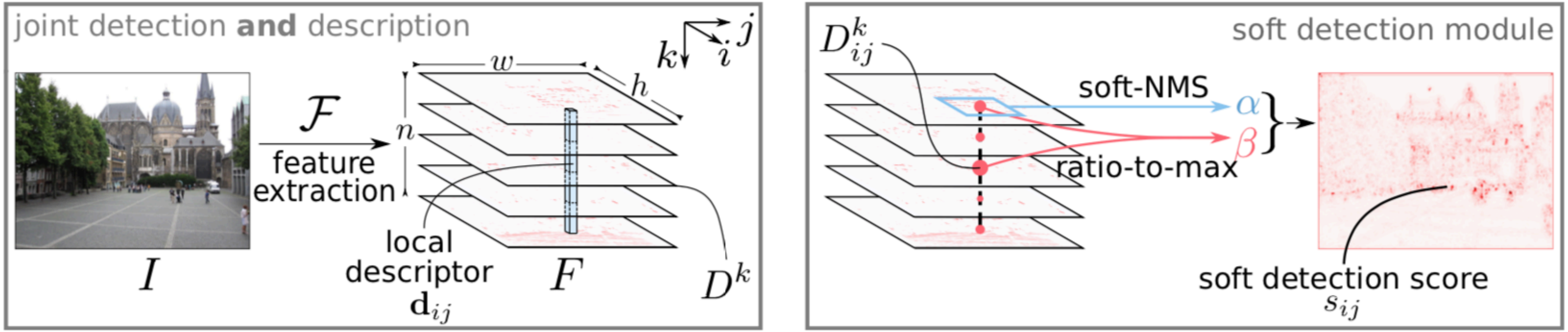


Figure 3: **Proposed detect-and-describe (D2) network.** A feature extraction CNN $\mathcal{F}$ is used to extract feature maps that play a dual role: (i) local descriptors $\mathbf{d}_{ij}$ are simply obtained by traversing all the n feature maps D^k at a spatial position (i, j); (ii) detections are obtained by performing a non-local-maximum suppression on a feature map followed by a non-maximum suppression across each descriptor - during training, keypoint detection scores s_{ij} are computed from a soft local-maximum score α and a ratio-to-maximum score per descriptor β.

Loss

- Descriptor + Detector
  - Weighted triplet loss

$$m(c) = \max\left(0, M + p(c)^2 - n(c)^2\right) \ . \qquad (12)$$

$$\mathcal{L}(I_1, I_2) = \sum_{c \in \mathcal{C}} \frac{s_c^{(1)} s_c^{(2)}}{\sum_{q \in \mathcal{C}} s_q^{(1)} s_q^{(2)}} m(p(c), n(c)) \ , \qquad (13)$$

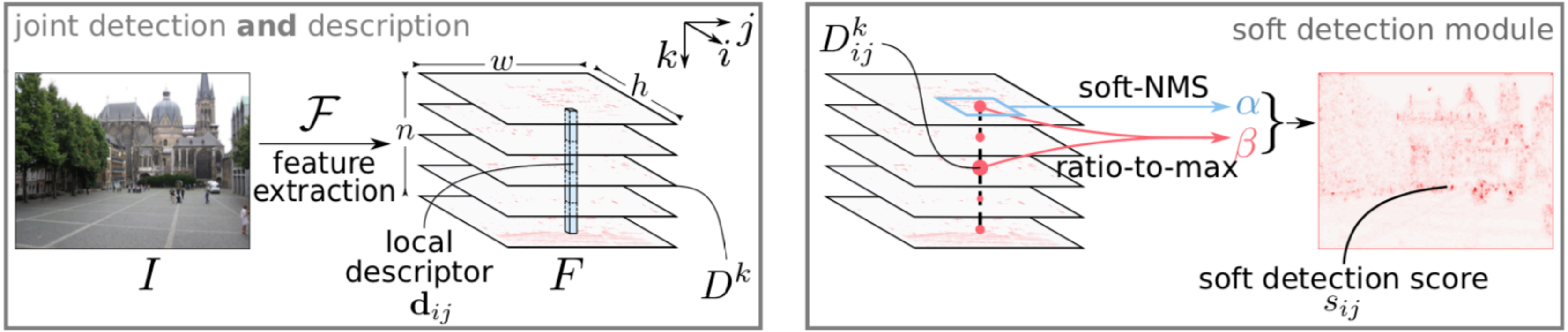
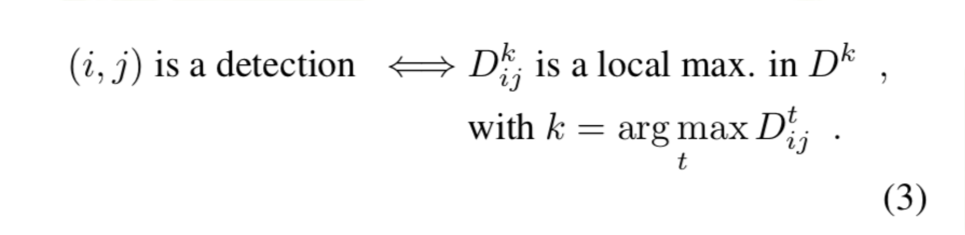


Figure 3: **Proposed detect-and-describe (D2) network.** A feature extraction CNN $\mathcal{F}$ is used to extract feature maps that play a dual role: (i) local descriptors $\mathbf{d}_{ij}$ are simply obtained by traversing all the n feature maps D^k at a spatial position (i, j); (ii) detections are obtained by performing a non-local-maximum suppression on a feature map followed by a non-maximum suppression across each descriptor - during training, keypoint detection scores s_{ij} are computed from a soft local-maximum score α and a ratio-to-maximum score per descriptor β.

- In traditional feature detectors such as DoG, the detection map would be sparsified by performing a spatial non-local-maximum suppression.
- However, in our approach, contrary to traditional feature detectors, there exist multiple detection maps

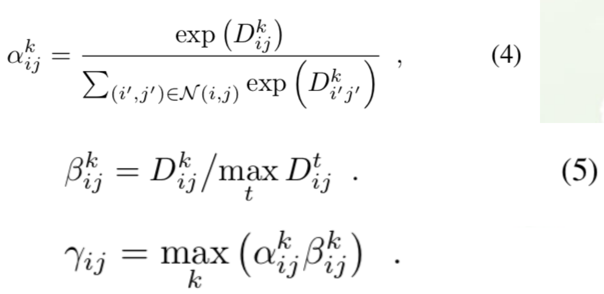
Get feature points

- Hard feature detection



$$(i,j) \text{ is a detection} \iff D^k_{ij} \text{ is a local max. in } D^k \ , \quad \text{with } k = \arg\max_t D^t_{ij} \ . \tag{3}$$

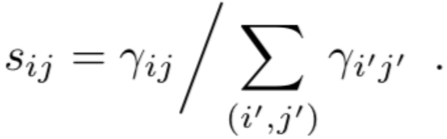
- Soft feature detection.



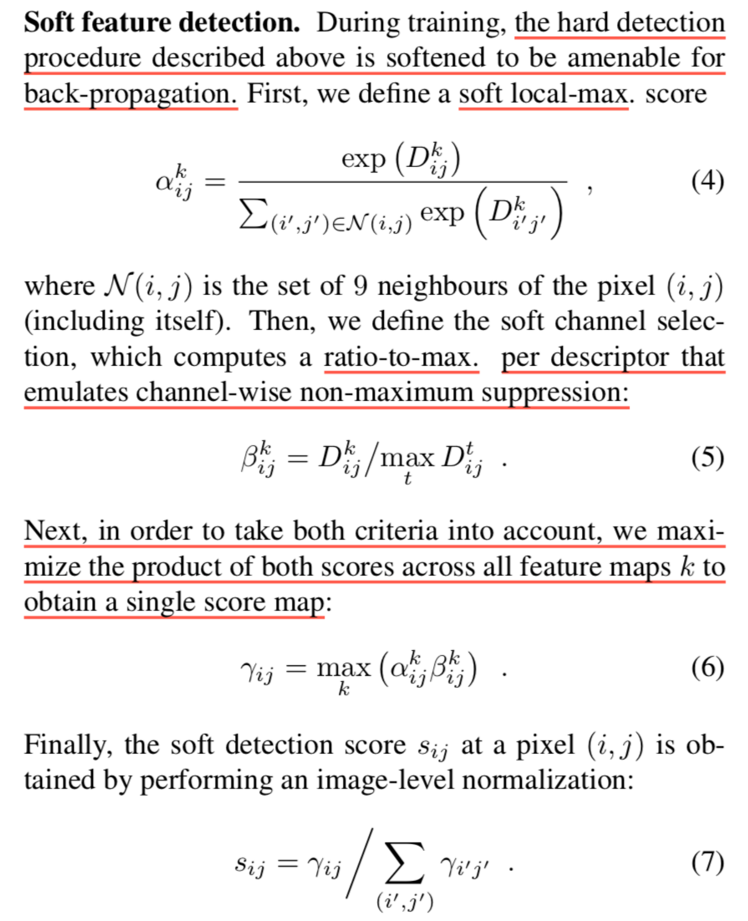
$$\alpha^k_{ij} = \frac{\exp\left(D^k_{ij}\right)}{\sum_{(i',j')\in\mathcal{N}(i,j)} \exp\left(D^k_{i'j'}\right)} \ , \tag{4}$$

$$\beta^k_{ij} = D^k_{ij} / \max_t D^t_{ij} \ . \tag{5}$$

$$\gamma_{ij} = \max_k \left(\alpha^k_{ij}\beta^k_{ij}\right) \ .$$



$$s_{ij} = \gamma_{ij} \Big/ \sum_{(i',j')} \gamma_{i'j'} \ .$$



Soft feature detection. During training, the hard detection procedure described above is softened to be amenable for back-propagation. First, we define a soft local-max. score

$$\alpha_{ij}^{k} = \frac{\exp\left(D_{ij}^{k}\right)}{\sum_{(i',j')\in\mathcal{N}(i,j)} \exp\left(D_{i'j'}^{k}\right)} \ , \tag{4}$$

where $\mathcal{N}(i,j)$ is the set of 9 neighbours of the pixel (i,j) (including itself). Then, we define the soft channel selection, which computes a ratio-to-max. per descriptor that emulates channel-wise non-maximum suppression:

$$\beta_{ij}^{k} = D_{ij}^{k} / \max_{t} D_{ij}^{t} \ . \tag{5}$$

Next, in order to take both criteria into account, we maximize the product of both scores across all feature maps k to obtain a single score map:

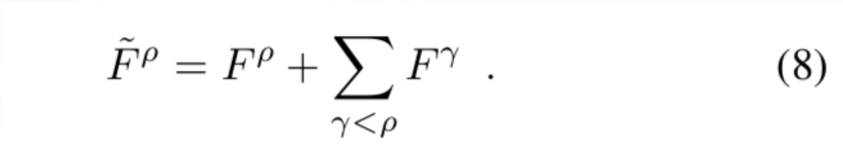
$$\gamma_{ij} = \max_{k} \left(\alpha_{ij}^{k} \beta_{ij}^{k}\right) \ . \tag{6}$$

Finally, the soft detection score s_{ij} at a pixel (i,j) is obtained by performing an image-level normalization:

$$s_{ij} = \gamma_{ij} \Big/ \sum_{(i',j')} \gamma_{i'j'} \ . \tag{7}$$

Multiscale detection at test time

- Multiscale Detection
  - we propose to use an image pyramid
  - This is only performed during test time.
  - ro = 0.5, 1, 2



$$\tilde{F}^{\rho} = F^{\rho} + \sum_{\gamma<\rho} F^{\gamma} \quad . \qquad (8)$$

  - Note that the feature maps F_ro have different resolutions.

Method details and analysis

- Loss
  - Descriptor: triplet margin ranking loss

$$m(c) = \max\left(0, M + p(c)^2 - n(c)^2\right) \quad . \qquad (12)$$

  - Descriptor + Detector:
    - weighted descriptor loss

$$\mathcal{L}(I_1, I_2) = \sum_{c \in \mathcal{C}} \frac{s_c^{(1)} s_c^{(2)}}{\sum_{q \in \mathcal{C}} s_q^{(1)} s_q^{(2)}} m(p(c), n(c)) \ , \qquad (13)$$

Method details and analysis

- Loss
  - Descriptor: triplet margin ranking loss

$$m(c) = \max\left(0, M + p(c)^2 - n(c)^2\right) \quad . \qquad (12)$$

  - Descriptor + Detector:
    - The proposed loss produces a weighted average of the margin terms m over all matches based on their detection scores.
    - Thus, in order for the loss to be minimized, the most distinctive correspondences (with a lower margin term) will get higher relative scores and vice-versa

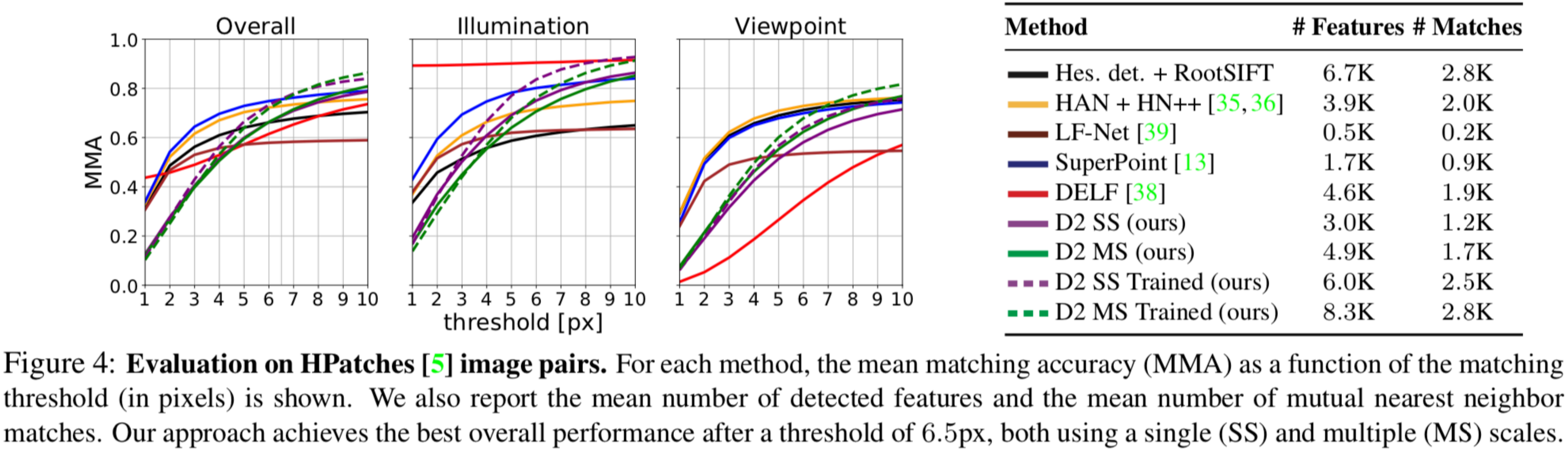
$$\mathcal{L}(I_1, I_2) = \sum_{c \in \mathcal{C}} \frac{s_c^{(1)} s_c^{(2)}}{\sum_{q \in \mathcal{C}} s_q^{(1)} s_q^{(2)}} m(p(c), n(c)) \ , \qquad (13)$$

Training Data

- Training Data
  - MegaDepth dataset consisting of 196 different scenes reconstructed from 1,070,468 internet photos using COLMAP

Experiments

- Image Matching



Method	# Features	# Matches
━ Hes. det. + RootSIFT	6.7K	2.8K
━ HAN + HN++ [35, 36]	3.9K	2.0K
━ LF-Net [39]	0.5K	0.2K
━ SuperPoint [13]	1.7K	0.9K
━ DELF [38]	4.6K	1.9K
━ D2 SS (ours)	3.0K	1.2K
━ D2 MS (ours)	4.9K	1.7K
╍ D2 SS Trained (ours)	6.0K	2.5K
╍ D2 MS Trained (ours)	8.3K	2.8K

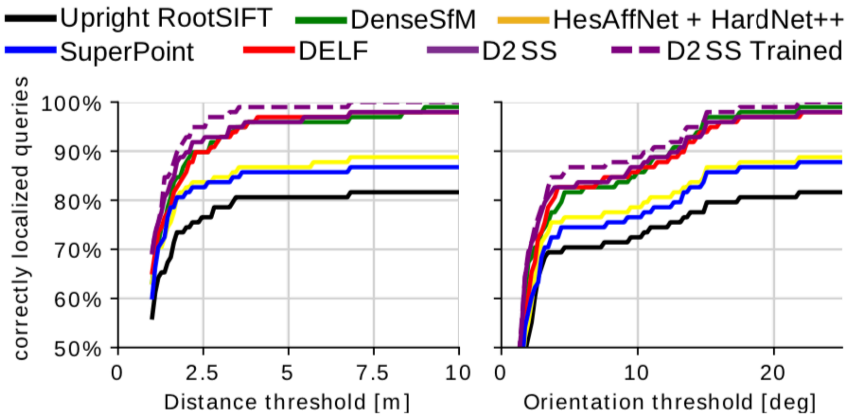
Figure 4: **Evaluation on HPatches [5] image pairs.** For each method, the mean matching accuracy (MMA) as a function of the matching threshold (in pixels) is shown. We also report the mean number of detected features and the mean number of mutual nearest neighbor matches. Our approach achieves the best overall performance after a threshold of 6.5px, both using a single (SS) and multiple (MS) scales.

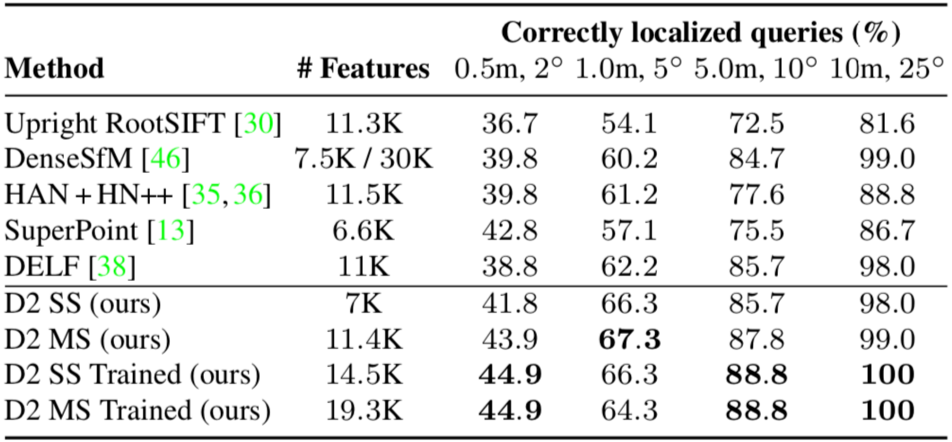
  - Worse for stricter matching threshold

- 3D Reconstruction
  - MVS

Experiments

- Localization under challenging Conditions
  - Day-Night Visual Localization





Method	# Features	Correctly localized queries (%)			
		0.5m, 2°	1.0m, 5°	5.0m, 10°	10m, 25°
Upright RootSIFT [30]	11.3K	36.7	54.1	72.5	81.6
DenseSfM [46]	7.5K / 30K	39.8	60.2	84.7	99.0
HAN + HN++ [35, 36]	11.5K	39.8	61.2	77.6	88.8
SuperPoint [13]	6.6K	42.8	57.1	75.5	86.7
DELF [38]	11K	38.8	62.2	85.7	98.0
D2 SS (ours)	7K	41.8	66.3	85.7	98.0
D2 MS (ours)	11.4K	43.9	**67.3**	87.8	99.0
D2 SS Trained (ours)	14.5K	**44.9**	66.3	**88.8**	**100**
D2 MS Trained (ours)	19.3K	**44.9**	64.3	**88.8**	**100**

Figure 5: **Evaluation on the Aachen Day-Night dataset [46, 48].** We report the percentage of images registered within given error thresholds. Our approach improves upon state-of-the art methods by a significant margin under strict pose thresholds.

Benchmarking 6dof outdoor visual localization in changing conditions

T Sattler, W Maddern, C Toft, A Torii, L Hammarstrand… - Proceedings of the IEEE …, 2018, CVPR(SPOT)

Qualitative results

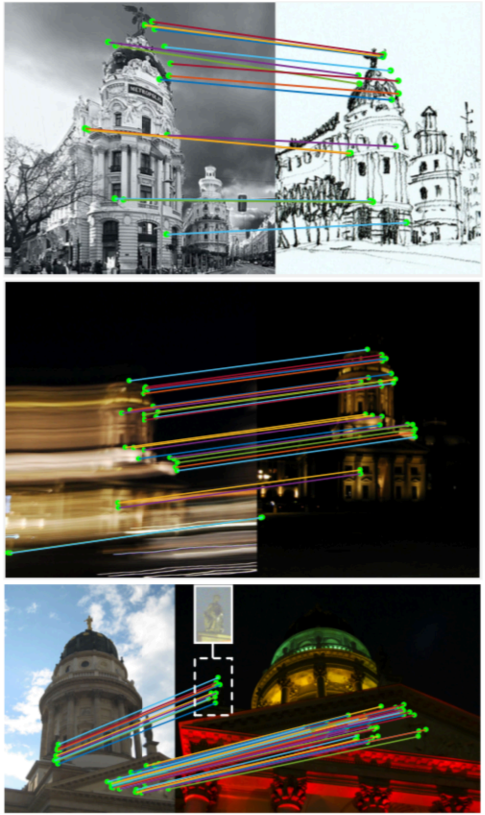
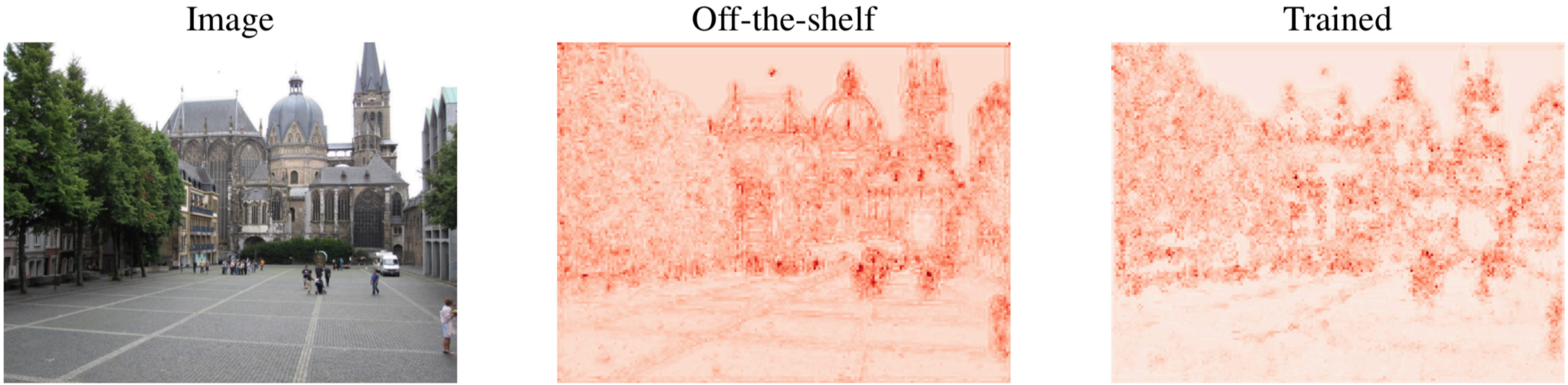


Figure 1: **Examples of matches obtained by the D2-Net method.** The proposed method can find image correspondences even under significant appearance differences caused by strong changes in illumination such as day-to-night, changes in depiction style or under image degradation caused by motion blur.

Qualitative results

- Soft detection scores for different scenes



Qualitative results - InLoc

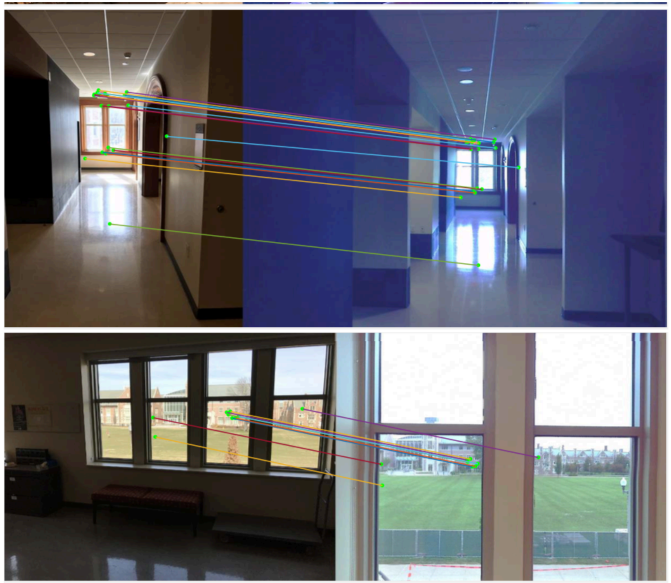


Figure 10: **Examples of correctly matched image pairs from the InLoc [59] dataset.** Our features are robust to significant changes in viewpoint as it can be seen in the first example. In textureless areas, our features act as an object matcher - correspondences are found between the furniture of different scenes. Sometimes, matches are even found across windows on nearby buildings.

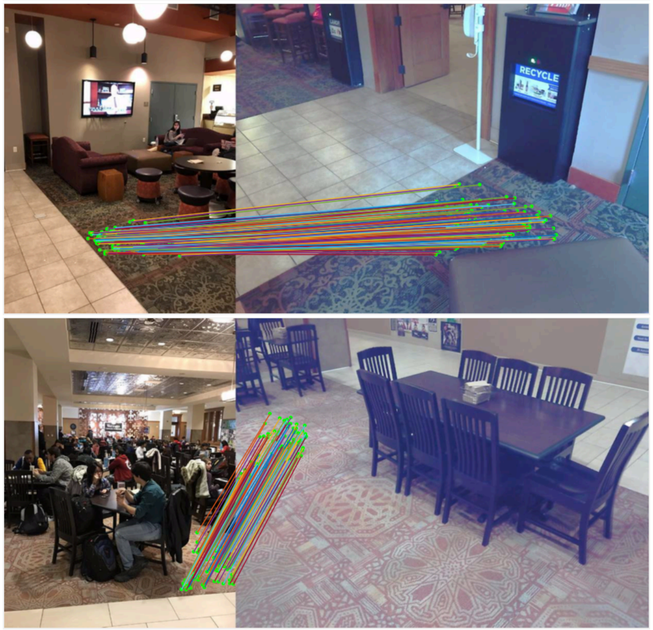


Figure 11: **Failure cases from the InLoc [59] dataset.** Even though they are visually correct, the matches sometimes put in correspondence identical objects from different scenes. Another typical error case is due to repeated patterns (e.g. on carpets) which yield a significant number of inliers.

Qualitative results – Day-Night

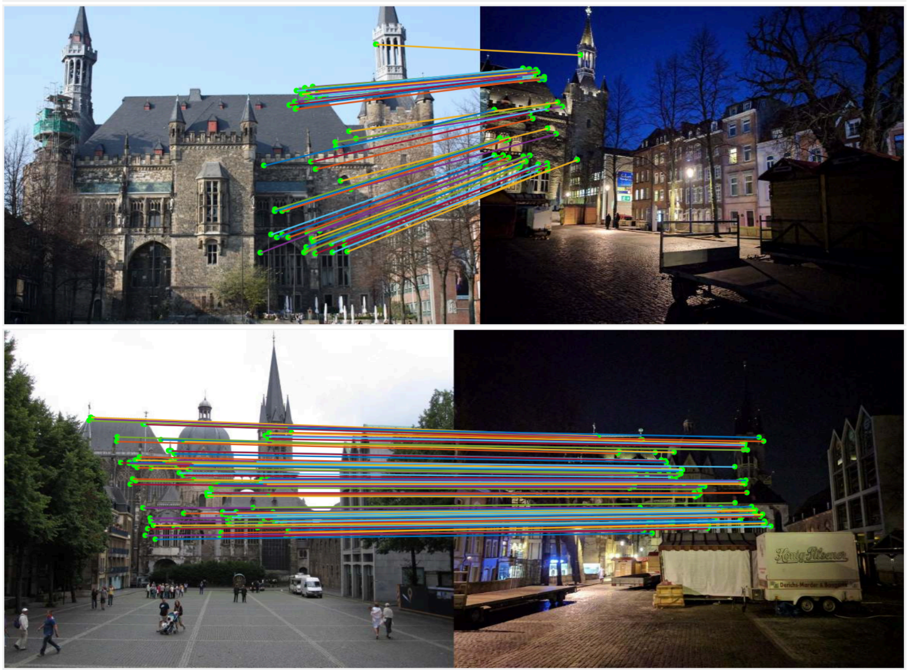


Figure 12: **Examples of correctly matched image pairs from the Aachen Day-Night [46,48] dataset.** Our features consistently provide a significant number of good matches between images with strong illumination changes. The first two image pairs come from scenes where no other method was able to register the night-time image. For the last two, DELF [38] was the only other method that succeeded.

In short

- D2-Net is deep learning approach for detector and descriptor
- One representation for both detector and descriptor
- Use outputs from off-the-shelf SFM method as supervision
- Weighted triplet loss

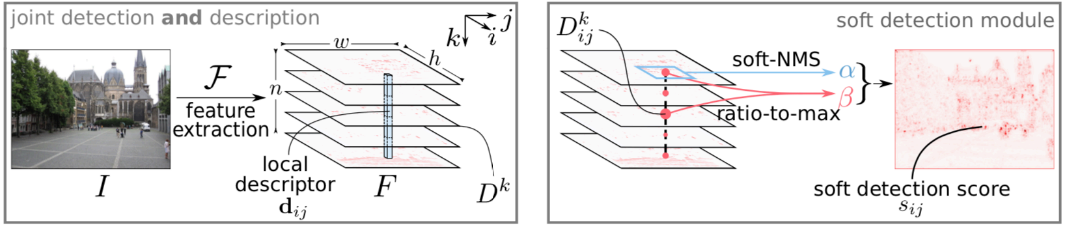


Figure 3: **Proposed detect-and-describe (D2) network.** A feature extraction CNN $\mathcal{F}$ is used to extract feature maps that play a dual role: (i) local descriptors $\mathbf{d}_{ij}$ are simply obtained by traversing all the n feature maps D^k at a spatial position (i, j); (ii) detections are obtained by performing a non-local-maximum suppression on a feature map followed by a non-maximum suppression across each descriptor - during training, keypoint detection scores s_{ij} are computed from a soft local-maximum score α and a ratio-to-maximum score per descriptor β.

Future work and discussion

- Feature points are not accurate
- Keep the resolution of image features
  - low resolution (1/8) when training

Questions?

ContextDesc: Local Descriptor Augmentation with Cross-Modality Context

Zixin Luo[1] Tianwei Shen[1] Lei Zhou[1] Jiahui Zhang[2]
Yao Yao[1] Shiwei Li[1] Tian Fang[3] Long Quan[1]
[1]Hong Kong University of Science and Technology
[2]Tsinghua University [3]Shenzhen Zhuke Innovation Technology (Altizure)
{zluoag,tshenaa,lzhouai,yyaoag,slibc,quan}@cse.ust.hk
jiahui-z15@mails.tsinghua.edu.cn fangtian@altizure.com

CVPR 2019 (oral)

ContextDesc: Local Descriptor Augmentation with Cross-Modality Context

Z Luo, T Shen, L Zhou, J Zhang, Y Yao, S Li, T Fang… - arXiv preprint arXiv …, 2019

Motivation and problem description

- Why local feature descriptor?
  - panorama stitching, wide-baseline matching, image retrieval and structure-from-motion (SfM)
- What is the challenge?
  - repetitive patterns
  - visually indistinguishable from ground truth
- What is the goal?
  - feature description with extra prior knowledge
  - effectively combine the local feature description and off-the-shelf visual understandings

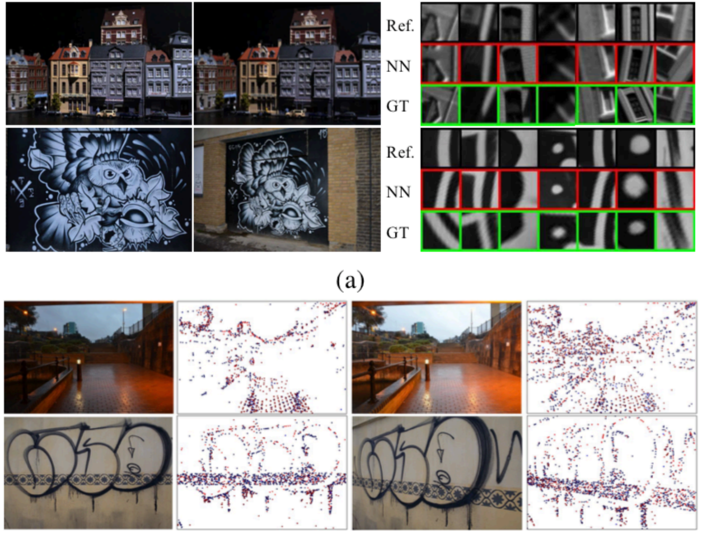


Figure 1: (a) Saturated results on standard benchmark [2] by a recent method [23]. The search of nearest neighbors (NN) returns false matches though visually similar to groundtruth (GT), indicating the limitation of relying on only local visual information. (b) 2D keypoints distribute structurally, on which we human beings are capable of establishing coarse matches even without color information.

Motivation and problem description

- Why local feature descriptor?
  - panorama stitch- ing [21], wide-baseline matching [24, 54, 55], image re- trieval [27] and structure-from-motion (SfM)
- What is the challenge?
  - due to repetitive patterns, the matching algorithm often finds false matches as nearest neighbors that are vi- sually indistinguishable from groundtruth
- What is the goal?
  - we seek to enhance the local feature description with extra prior knowledge, which we refer to as introducing context awareness to augment local feature descriptors.
  - Previously, a multi-scale-like architecture can help to capture visual context of different levels
  - we strive to effectively combine the local feature description and off-the-shelf visual understandings so as to go beyond the local detail representation.

Contributions

- A novel visual context encoder
  - high-level visual understandings
- A novel geometric context encoder
  - unordered points and exploits geometric cues
- A novel N-pair loss
  - requires no manual hyper-parameter search
  - better convergence properties.

Contributions

- Contributions
  - a novel visual context encoder that integrates high-level visual understandings from regional image representation, a technique often used by image retrieval
  - A novel geometric context encoder that consumes unordered points and exploits geometric cues from 2D keypoint distri-bution, while being robust to complex variations.
  - A novel N-pair loss that requires no manual hyper-parameter search and has better convergence properties.

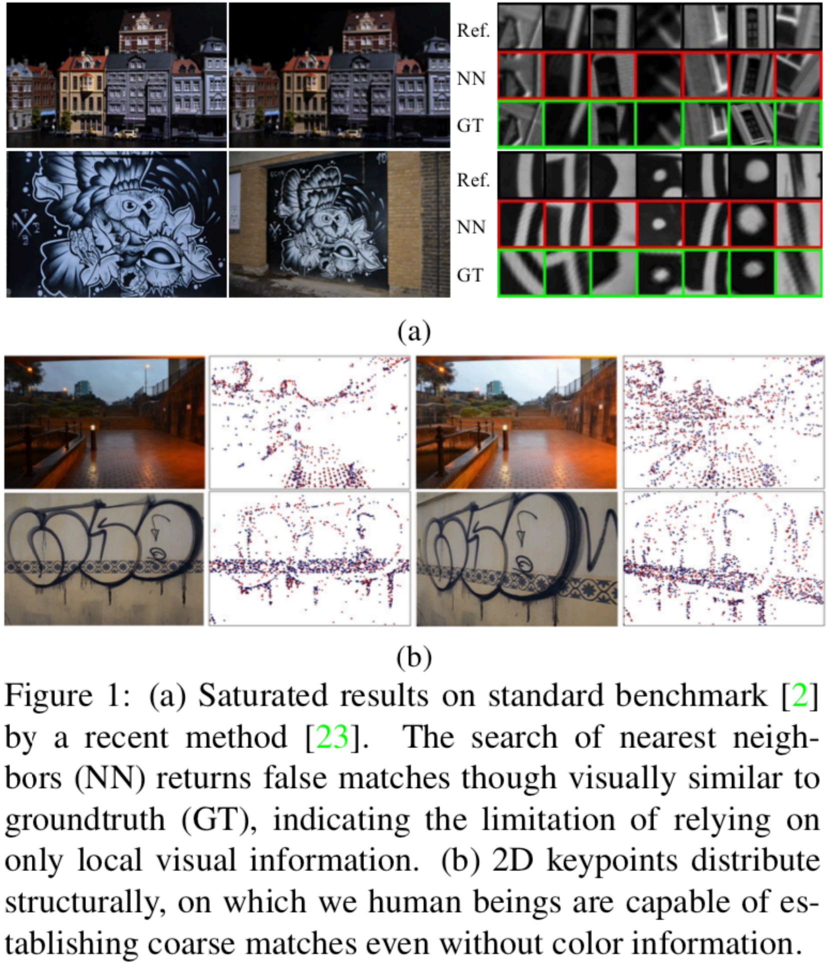


Figure 1: (a) Saturated results on standard benchmark [2] by a recent method [23]. The search of nearest neighbors (NN) returns false matches though visually similar to groundtruth (GT), indicating the limitation of relying on only local visual information. (b) 2D keypoints distribute structurally, on which we human beings are capable of establishing coarse matches even without color information.

Prior work

- Learned local descriptors
  - take individual image patches as input

Medium: LSD-slam and ORB-slam2, a literature based explanation

Method overview

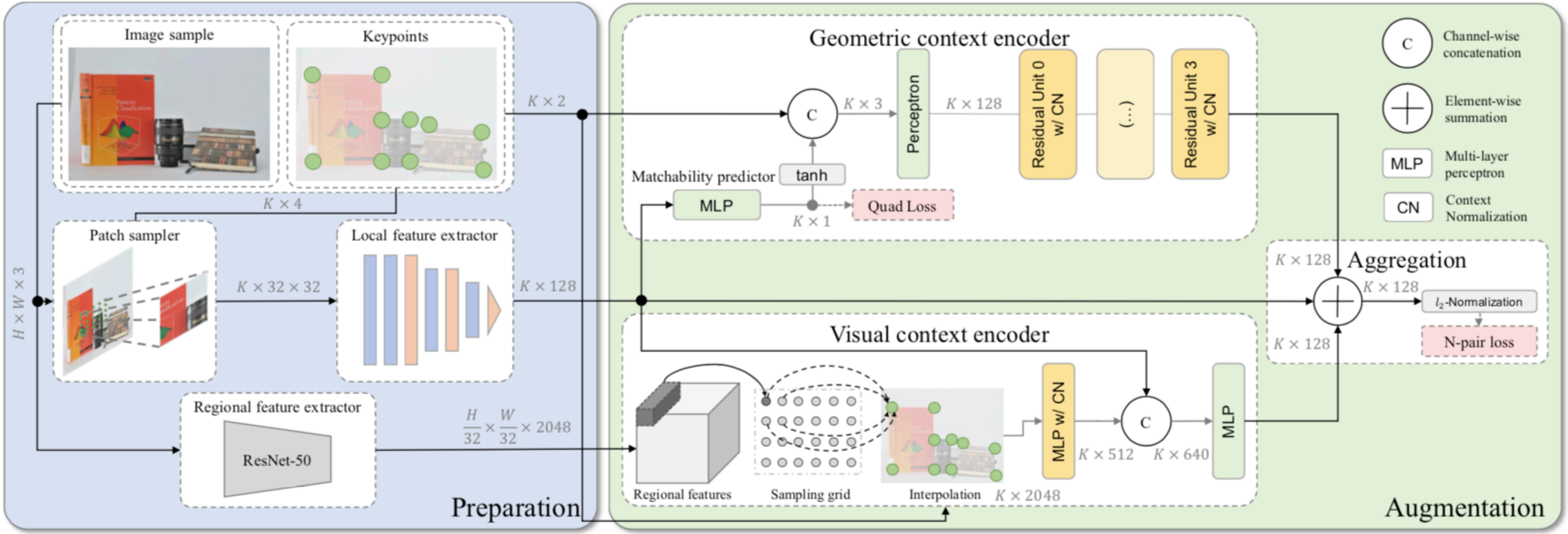


Figure 2: The proposed augmentation framework consumes a single image as input, from which 2D keypoints, local and regional features are extracted and encoded as geometric and visual context to improve the raw local feature description.

Preparation module

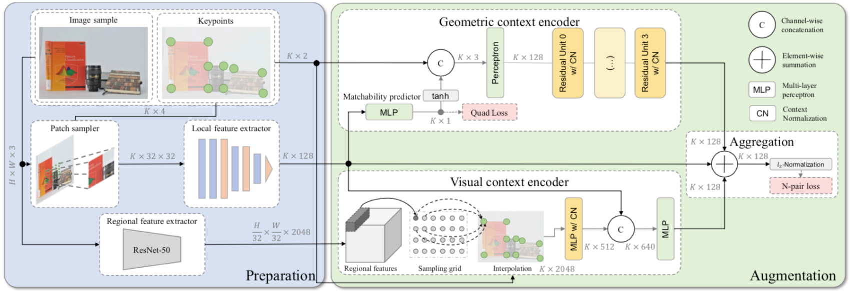


Figure 2: The proposed augmentation framework consumes a single image as input, from which 2D keypoints, local and regional features are extracted and encoded as geometric and visual context to improve the raw local feature description.

- Patch sampler
  - SIFT feature detector
  - 32 x 32 gray-scale patches
  - sampled by a spatial transformer from SIFT
- Local feature extractor
  - takes image patches as input, producing 128-d feature descriptions as output
  - 7-layer CNN
- Regional feature extractor
  - features from an off-the-shelf deep image retrieval model of ResNet-50

Visual context encoder

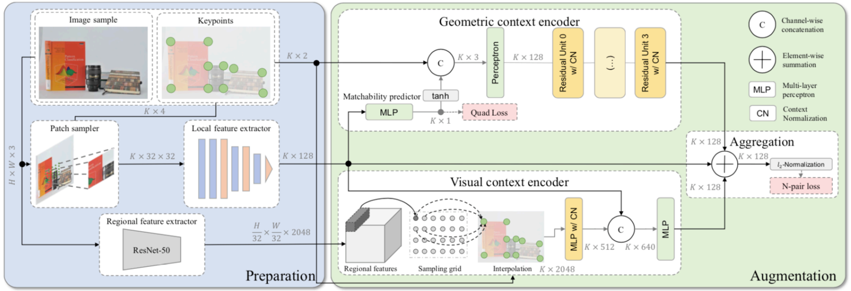


Figure 2: The proposed augmentation framework consumes a single image as input, from which 2D keypoints, local and regional features are extracted and encoded as geometric and visual context to improve the raw local feature description.

- kNN interpolation
- concatenate raw local features and regional features
- Pass through MLPs, forming the final 128-d features.

Geometric context encoder

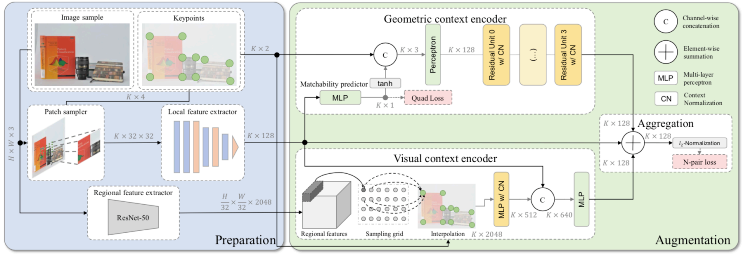
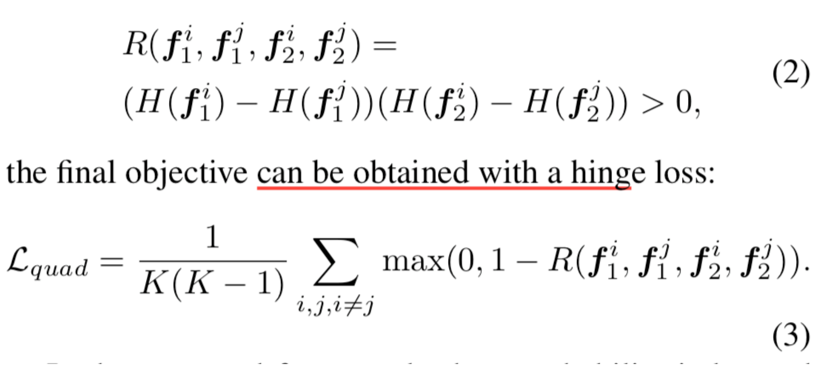


Figure 2: The proposed augmentation framework consumes a single image as input, from which 2D keypoints, local and regional features are extracted and encoded as geometric and visual context to improve the raw local feature description.

- This module takes K unordered points as input, and outputs 128-d corresponding feature vectors.
  - point coordinates
  - Matchability predictor



$$R(\boldsymbol{f}_1^i, \boldsymbol{f}_1^j, \boldsymbol{f}_2^i, \boldsymbol{f}_2^j) = (H(\boldsymbol{f}_1^i) - H(\boldsymbol{f}_1^j))(H(\boldsymbol{f}_2^i) - H(\boldsymbol{f}_2^j)) > 0, \tag{2}$$

the final objective can be obtained with a hinge loss:

$$\mathcal{L}_{quad} = \frac{1}{K(K-1)} \sum_{i,j,i \neq j} \max(0, 1 - R(\boldsymbol{f}_1^i, \boldsymbol{f}_1^j, \boldsymbol{f}_2^i, \boldsymbol{f}_2^j)). \tag{3}$$

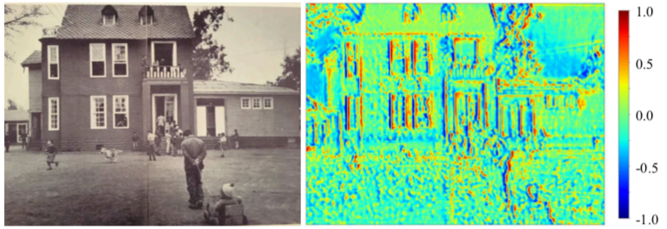
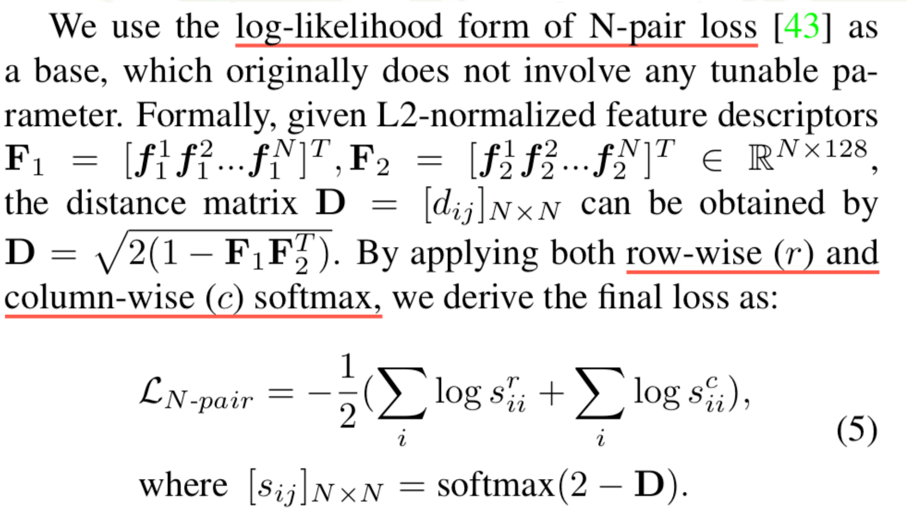


Figure 4: Visualization of matchability responding to the entire image (best viewed in color).

Loss

- N-pair loss
  - positive pairs closer
  - negative pairs farther



We use the log-likelihood form of N-pair loss [43] as a base, which originally does not involve any tunable parameter. Formally, given L2-normalized feature descriptors $\mathbf{F}_1 = [\boldsymbol{f}_1^1 \boldsymbol{f}_1^2 ... \boldsymbol{f}_1^N]^T, \mathbf{F}_2 = [\boldsymbol{f}_2^1 \boldsymbol{f}_2^2 ... \boldsymbol{f}_2^N]^T \in \mathbb{R}^{N \times 128}$, the distance matrix $\mathbf{D} = [d_{ij}]_{N \times N}$ can be obtained by $\mathbf{D} = \sqrt{2(1 - \mathbf{F}_1 \mathbf{F}_2^T)}$. By applying both row-wise (r) and column-wise (c) softmax, we derive the final loss as:

$$\mathcal{L}_{N\text{-}pair} = -\frac{1}{2}(\sum_i \log s_{ii}^r + \sum_i \log s_{ii}^c), \tag{5}$$

where $[s_{ij}]_{N \times N} = \text{softmax}(2 - \mathbf{D})$.

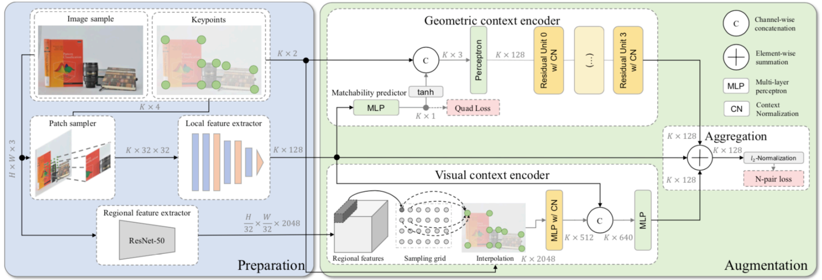
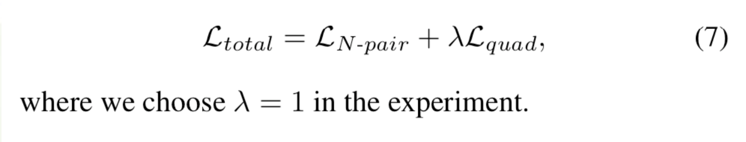


Figure 2: The proposed augmentation framework consumes a single image as input, from which 2D keypoints, local and regional features are extracted and encoded as geometric and visual context to improve the raw local feature description.



$$\mathcal{L}_{total} = \mathcal{L}_{N\text{-}pair} + \lambda \mathcal{L}_{quad}, \tag{7}$$

where we choose $\lambda = 1$ in the experiment.

Feature aggregation

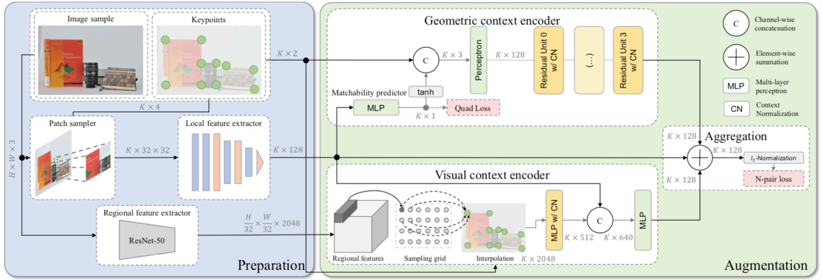


Figure 2: The proposed augmentation framework consumes a single image as input, from which 2D keypoints, local and regional features are extracted and encoded as geometric and visual context to improve the raw local feature description.

- element-wise summation and L2-normalization
  - same feature dimensionality
- Flexible
  - still work when only geometric context is available

Method overview

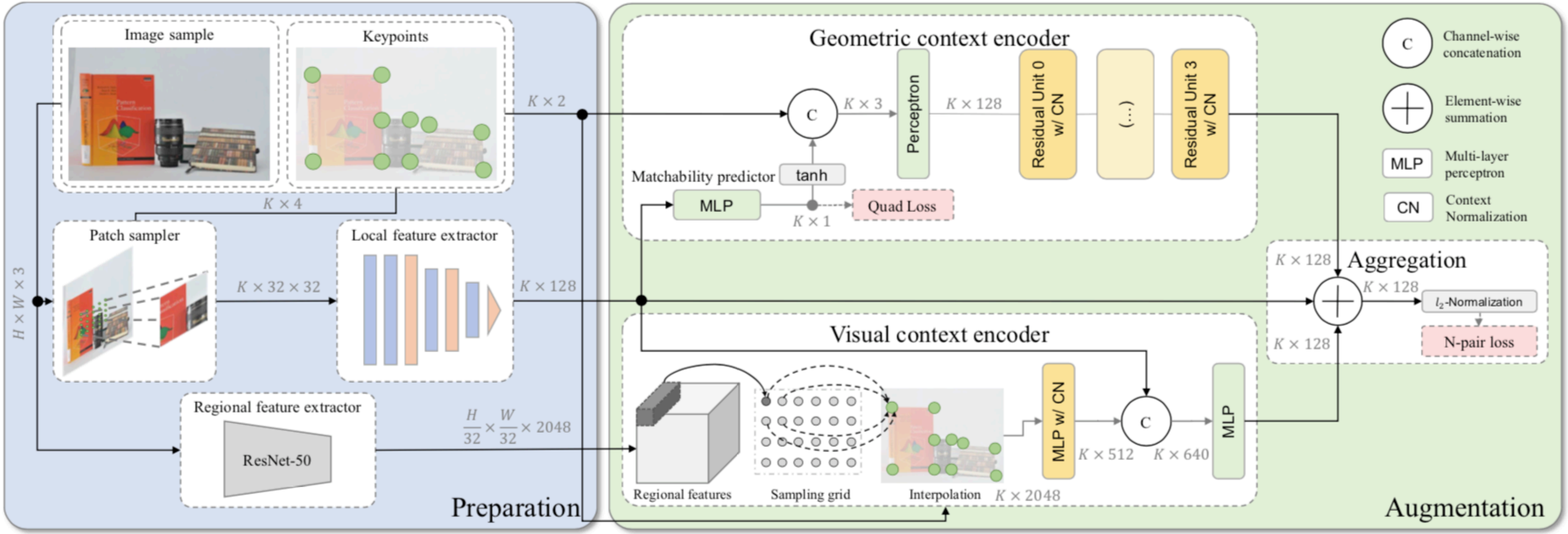


Figure 2: The proposed augmentation framework consumes a single image as input, from which 2D keypoints, local and regional features are extracted and encoded as geometric and visual context to improve the raw local feature description.

Visual context encoder

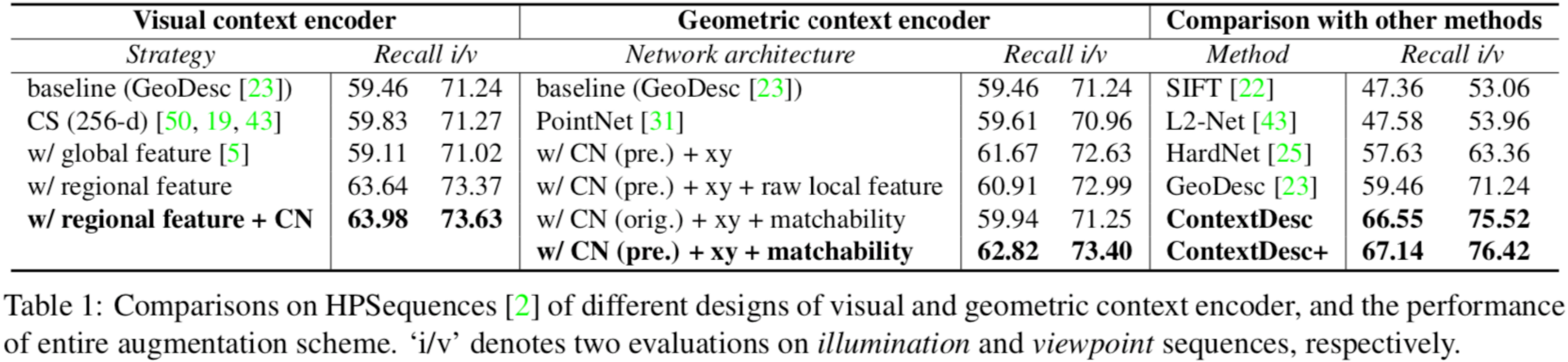
- In our framework, the global feature can be derived by applying Maximum Activations of Convolutions (MAC) aggregation, which simply max-pools over all dimensions of regional features.
- kNN interpolation
- Finally, raw local features are concatenated and further mapped by MLPs, forming the final 128-d features.

Implementation details

- Training dataset.
  - large-scale photo-tourism and aerial datasets (GL3D), and generate ground truth matches from SfM
- Data augmentation

Experiments

- Matchings on Hpatches
  - Recall = # Correct Matches / # Correspondences



Visual context encoder			Geometric context encoder			Comparison with other methods		
Strategy	*Recall i/v*		*Network architecture*	*Recall i/v*		*Method*	*Recall i/v*	
baseline (GeoDesc [23])	59.46	71.24	baseline (GeoDesc [23])	59.46	71.24	SIFT [22]	47.36	53.06
CS (256-d) [50, 19, 43]	59.83	71.27	PointNet [31]	59.61	70.96	L2-Net [43]	47.58	53.96
w/ global feature [5]	59.11	71.02	w/ CN (pre.) + xy	61.67	72.63	HardNet [25]	57.63	63.36
w/ regional feature	63.64	73.37	w/ CN (pre.) + xy + raw local feature	60.91	72.99	GeoDesc [23]	59.46	71.24
w/ regional feature + CN	**63.98**	**73.63**	w/ CN (orig.) + xy + matchability	59.94	71.25	**ContextDesc**	**66.55**	**75.52**
			w/ CN (pre.) + xy + matchability	**62.82**	**73.40**	**ContextDesc+**	**67.14**	**76.42**

Table 1: Comparisons on HPSequences [2] of different designs of visual and geometric context encoder, and the performance of entire augmentation scheme. ‘i/v’ denotes two evaluations on *illumination* and *viewpoint* sequences, respectively.

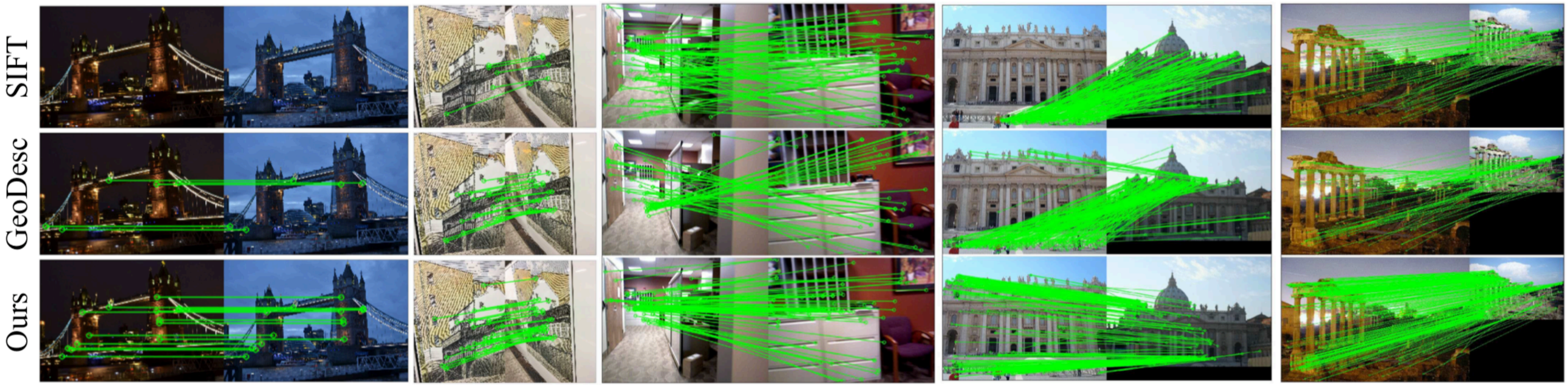


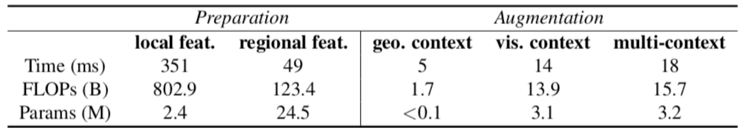
Figure 5: Matching results after RANSAC in different challenging scenarios. From top to bottom: SIFT, GeoDesc and ours. The augmented feature helps to find more inlier matches, and further allows a more accurate recovery of camera geometry.

In short

- Only descriptor
- Local patches (VGG-like) + global features (ResNet) + Point coordinates (PointNet)

Future work and discussion

- Computational cost
  - Slow



	Preparation		*Augmentation*		
	local feat.	**regional feat.**	**geo. context**	**vis. context**	**multi-context**
Time (ms)	351	49	5	14	18
FLOPs (B)	802.9	123.4	1.7	13.9	15.7
Params (M)	2.4	24.5	<0.1	3.1	3.2

Table 6: The computational cost of proposed framework, evaluated on 10k keypoints from an 896×896 image. The inference time is estimated on an NVIDIA GTX 1080 GPU.

- End-to-end training
  - we freeze only the regional model and train from scratch with Eq. 7 on the augmented feature
  - end-to-end train with the regional model. No improvement.

Questions?

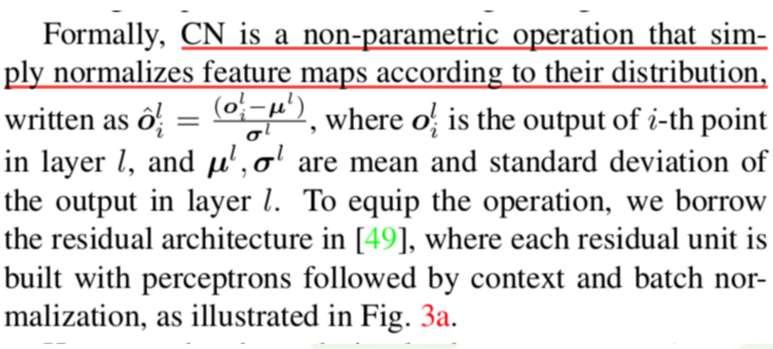
Takeaways

- SIFT is robust with subpixel accuracy
  - LF-Net: softargmax
  - Change backbone: receptive field vs. efficiency
- Training dataset is from COLMAP

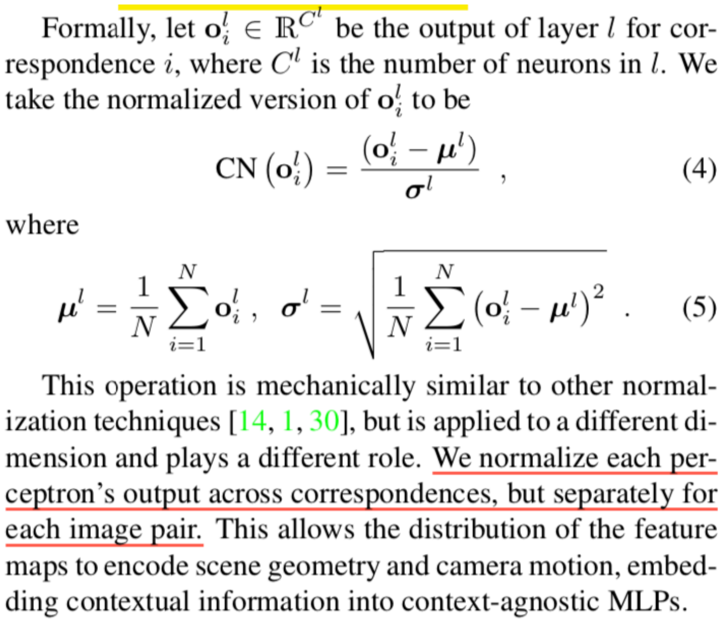
Thank you

Backup

Context normalization



Formally, CN is a non-parametric operation that simply normalizes feature maps according to their distribution, written as $\hat{\boldsymbol{o}}_i^l = \frac{(\boldsymbol{o}_i^l - \boldsymbol{\mu}^l)}{\boldsymbol{\sigma}^l}$, where $\boldsymbol{o}_i^l$ is the output of i-th point in layer l, and $\boldsymbol{\mu}^l, \boldsymbol{\sigma}^l$ are mean and standard deviation of the output in layer l. To equip the operation, we borrow the residual architecture in [49], where each residual unit is built with perceptrons followed by context and batch normalization, as illustrated in Fig. 3a.



Formally, let $\mathbf{o}_i^l \in \mathbb{R}^{C^l}$ be the output of layer l for correspondence i, where C^l is the number of neurons in l. We take the normalized version of $\mathbf{o}_i^l$ to be

$$\mathrm{CN}\left(\mathbf{o}_i^l\right) = \frac{\left(\mathbf{o}_i^l - \boldsymbol{\mu}^l\right)}{\boldsymbol{\sigma}^l} \quad , \tag{4}$$

where

$$\boldsymbol{\mu}^l = \frac{1}{N}\sum_{i=1}^{N} \mathbf{o}_i^l \ , \quad \boldsymbol{\sigma}^l = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\mathbf{o}_i^l - \boldsymbol{\mu}^l\right)^2} \quad . \tag{5}$$

This operation is mechanically similar to other normalization techniques [14, 1, 30], but is applied to a different dimension and plays a different role. We normalize each perceptron's output across correspondences, but separately for each image pair. This allows the distribution of the feature maps to encode scene geometry and camera motion, embedding contextual information into context-agnostic MLPs.

Learning to find good correspondences

KM Yi, E Trulls Fortuny, Y Ono, V Lepetit, M Salzmann… - Proceedings of the 2018 …, 2018