SOSNet: Second Order Similarity Regularization for Local Descriptor Learning

Yurun Tian*,† Xin Yu*,^ Bin Fan† Fuchao Wu† Huub Heijnen* Vassileios Balntas*

Scape Technologies, UK* CASIA, China† ANU, Australia^

CVPR JUNE 16-21, 2019 LONG BEACH • CALIFORNIA

Problem: Image Matching

Matching two images from different viewpoints, under changing conditions.

Keypoints → Patches → Descriptors

Challenges & Motivation

Challenges

- Keypoint repeatability
  - Design more robust keypoint detection methods
- Descriptor robustness
  - Design more robust patch description methods [SOSNet]

Motivation: Triplet Loss Inconsistency

before training → after training

Triplet loss does not enforce inter-class consistency

First & Second Order Similarities

- Previous work: optimise First Order Similarity (FOS)
  - Distances between descriptors
- This Work: Besides FOS, we can utilise Second Order Similarity (SOS) information to improve matching.
  - Distances between FOS distances
- FOS and SOS can be jointly optimized.

$$\mathcal{L} = \mathcal{L}_{\text{FOS}} + \mathcal{R}_{\text{SOS}}$$

Optimizing First Order Similarity

Quadratic Hinge Triplet (QHT) Loss + hard sample mining as in *HardNet*

$$\mathcal{L}_{\text{FOS}} = \frac{1}{N}\sum_{i=1}^{N} \max \underbrace{\boxed{(0, t + d_i^{\text{pos}} - d_i^{\text{neg}})^2}}_{\text{(QHT)}},$$

$$d_i^{\text{pos}} = d(\boldsymbol{x}_i, \boldsymbol{x}_i^+),$$

$$d_i^{\text{neg}} = \min_{\forall j, j \neq i} (d(\boldsymbol{x}_i, \boldsymbol{x}_j), d(\boldsymbol{x}_i, \boldsymbol{x}_j^+), d(\boldsymbol{x}_i^+, \boldsymbol{x}_j), d(\boldsymbol{x}_i^+, \boldsymbol{x}_j^+))$$

Optimizing Second Order Similarity

- Matching descriptors should have high SOS.

$$\mathcal{R}_{\text{SOS}} = \frac{1}{N}\sum_{i=1}^{N} d^{(2)}(\boldsymbol{x}_i, \boldsymbol{x}_i^+).$$

- We optimise second-order similarity by optimising the distance between distances.

$$\boldsymbol{y} = [\delta(\boldsymbol{x}, \boldsymbol{\alpha}), \delta(\boldsymbol{x}, \boldsymbol{\beta})]$$
$$\boldsymbol{y}^+ = [\delta(\boldsymbol{x}^+, \boldsymbol{\alpha}^+), \delta(\boldsymbol{x}^+, \boldsymbol{\beta}^+)]$$
$$d^{(2)}(\boldsymbol{x}, \boldsymbol{x}^+) = ||\boldsymbol{y} - \boldsymbol{y}^+||$$
$$\downarrow$$
$$d^{(2)}(\boldsymbol{x}, \boldsymbol{x}^+) = \sqrt{\{\delta(\boldsymbol{x}, \boldsymbol{\alpha}) - \delta(\boldsymbol{x}^+, \boldsymbol{\alpha}^+)\}^2 + \{\delta(\boldsymbol{x}, \boldsymbol{\beta}) - \delta(\boldsymbol{x}^+, \boldsymbol{\beta}^+)\}^2}$$

kNN search strategy for the selection of reference descriptors

$$\mathcal{R}_{\text{SOS}} = \frac{1}{N}\sum_{i=1}^{N} d^{(2)}(\boldsymbol{x}_i, \boldsymbol{x}_i^+).$$

$$d^{(2)}(\boldsymbol{x}_i, \boldsymbol{x}_i^+) = \sqrt{\sum_{j \neq i}^{N} (d(\boldsymbol{x}_i, \boldsymbol{x}_j) - d(\boldsymbol{x}_i^+, \boldsymbol{x}_j^+))^2}$$

$$\boldsymbol{c}_i = \{z_j : \boldsymbol{x}_i \in KNN(\boldsymbol{x}_j) \vee \boldsymbol{x}_i^+ \in KNN(\boldsymbol{x}_j^+)\},$$
$$\forall j \in 1 \ldots N, j \neq i$$

Benchmarks

Train	Notredame	Yosemite	Liberty	Yosemite	Liberty	Notredame	Mean
Test	Liberty		Notredame		Yosemite		
SIFT [29]	29.84		22.53		27.29		26.55
DeepDesc [32]	10.9		4.40		5.69		6.99
MatchNet [7]	7.04	11.47	3.82	5.65	11.6	8.70	8.05
L2Net [34]	3.64	5.29	1.15	1.62	4.43	3.30	3.24
CS L2Net [34]	2.55	4.24	0.87	1.39	3.81	2.84	2.61
HardNet [25]	1.47	**2.67**	0.62	0.88	2.14	1.65	1.57
HardNet-GOR [25, 39]	1.72	2.89	0.63	0.91	2.10	1.59	1.64
Michel *et al.* [16]	1.79	2.96	0.68	1.02	2.51	1.64	1.77
SOSNet	**1.25**	2.84	**0.58**	**0.87**	**1.95**	**1.25**	**1.46**
TFeat+ [2]	7.39	10.13	3.06	3.80	8.06	7.24	6.64
L2Net+ [34]	2.36	4.70	0.72	1.29	2.57	1.71	2.23
CS L2Net+ [34]	1.71	3.87	0.56	1.09	2.07	1.3	1.76
HardNet+ [25]	1.49	2.51	0.53	0.78	1.96	1.84	1.51
HardNet-GOR+ [25, 39]	1.48	2.43	0.51	0.78	1.76	1.53	1.41
DOAP+ [14]	1.54	2.62	0.43	0.87	2.00	1.21	1.45
DOAP-ST+ [14] [15]	1.47	2.29	0.39	0.78	1.98	1.35	1.38
SOSNet+	**1.08**	**2.12**	**0.35**	**0.67**	**1.03**	**0.95**	**1.03**
GeoDesc+ [21]	5.47		1.94		4.72		4.05
SOSNet-HP+	2.10		0.79		1.39		1.42

		# Image	# Registered	# Sparse Points	# Observations	Track Length	Reproj. Error
Fountain	SIFT	11	11	14K	70K	4.79	0.39px
	DSP-SIFT		11	14K	71K	4.78	0.37px
	L2Net		11	17K	83K	4.88	0.47px
	GeoDesc		11	16K	83K	5.00	0.47px
	SOSNet		11	17K	85K	4.92	0.43px
Herzjesu	SIFT	8	8	13K	31K	4.22	0.43px
	DSP-SIFT		8	7.7K	32K	4.22	0.45px
	L2Net		8	9.5K	40K	4.24	0.51px
	GeoDesc		8	9.7K	40K	4.35	0.51px
	SOSNet		8	9.7K	41K	4.26	0.53px
South Building	SIFT	128	128	108K	651K	6.04	0.56px
	DSP-SIFT		128	112K	666K	5.91	0.58px
	L2Net		128	170K	863K	5.07	0.63px
	GeoDesc		128	172K	882K	5.21	0.64px
	SOSNet		128	173K	915K	5.11	0.67px
Madrid Metropolis	SIFT	1344	500	110K	733K	6.32	0.66px
	DSP-SIFT		467	99K	649K	6.52	0.66px
	L2Net		692	254K	1067K	4.20	0.69px
	GeoDesc		806	306K	1260K	3.91	0.66px
	SOSNet		844	315K	1311K	4.21	0.70px
Gendarmenmarkt	SIFT	1463	1035	338K	1872K	5.523	0.69px
	DSP-SIFT		979	293K	1577K	5.381	0.74px
	L2Net		1168	667K	2611K	3.91	0.73px
	GeoDesc		1208	753K	2951K	3.72	0.74px
	SOSNet		1201	816K	3255K	3.984	0.77px

Patches Results

vMF Analysis of descriptor spaces

We define a von Mises-Fisher distribution performance measure (ρ)

Distribution of class centers	Distribution of elements inside a class
κ_{inter} ⇓	κ_{intra} ⇑

$$\rho = \frac{\kappa_{\text{inter}}}{\kappa_{\text{intra}}} \propto \frac{R_{\text{inter}}}{R_{\text{intra}}}$$