ICCV 2019 Seoul, Korea

Hyperpixel Flow: Semantic Correspondence with Multi-layer Neural Features

[1,2]Juhong Min [1,2]Jongmin Lee [3,4]Jean Ponce [1,2]Minsu Cho
[1]POSTECH [2]NPRC [3]Inria [4]DI ENS
POSTECH POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY
Inria inventeurs du monde numérique

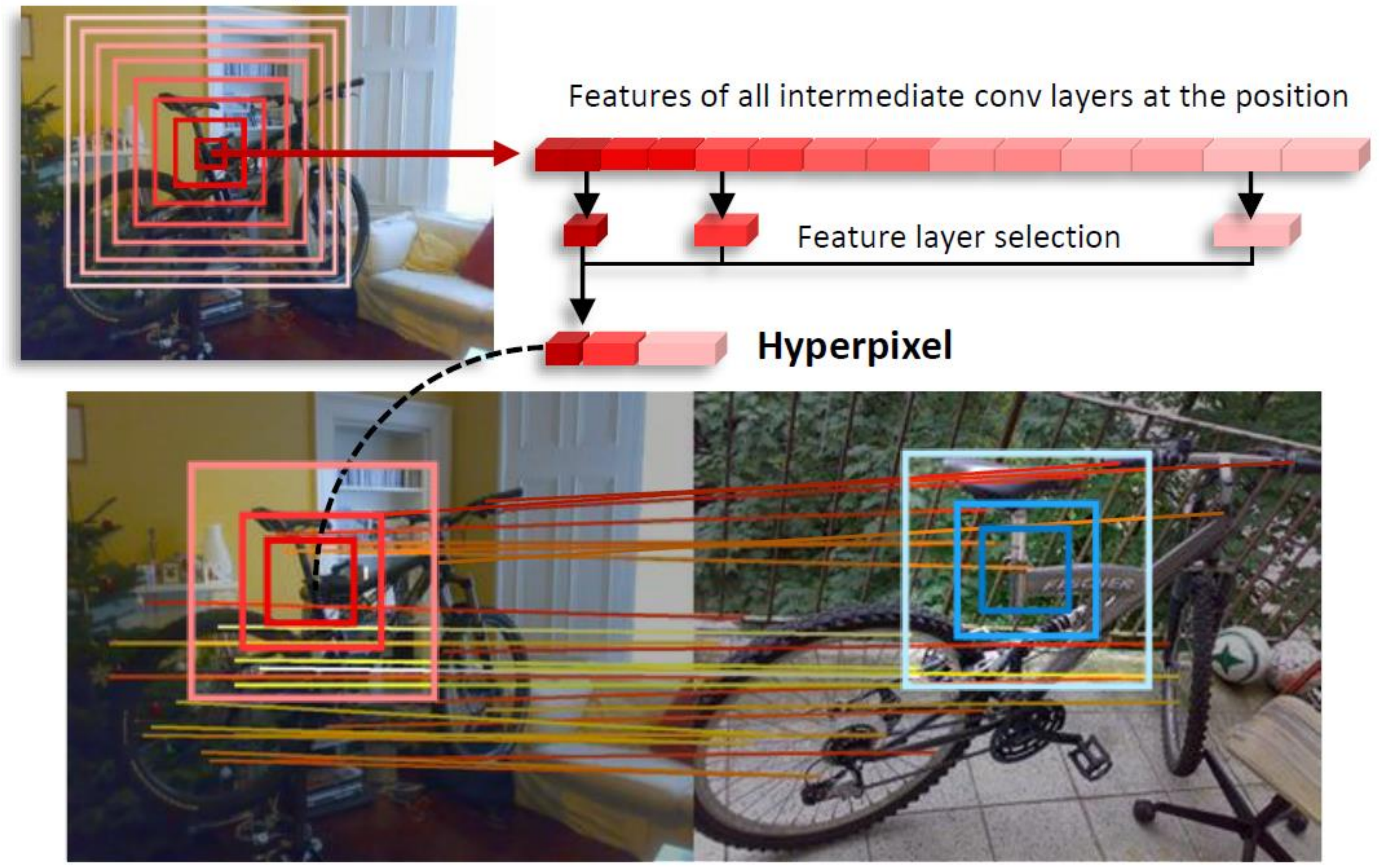
Hyperpixel Flow project page
SPair-71k project page
Code (GitHub)

Problem definition and motivation

- **Semantic correspondence:**
  Matching images depicting different instances of the same class
- **Limitation of existing approaches:**
  Prediction relies on features from a specific convolutional layer
  Fails to fully exploit different levels of semantic features
- **Limitation of existing datasets:**
  Small number of image pairs with similar viewpoints and scales
  Limited annotation types and no clear splits for learning

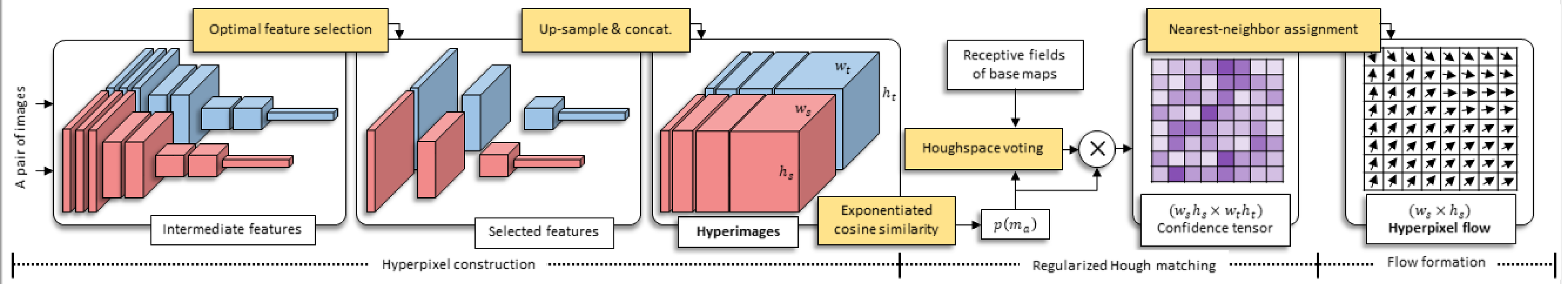
Contributions

1. Establish reliable correspondences using **multi-layer features**
2. Propose an **efficient, real-time** matching framework
3. SOTA using only a **small number of validation pairs** for model tuning
4. Introduce a **large-scale dataset** with richer annotations



Proposed method

- **Overall architecture:**

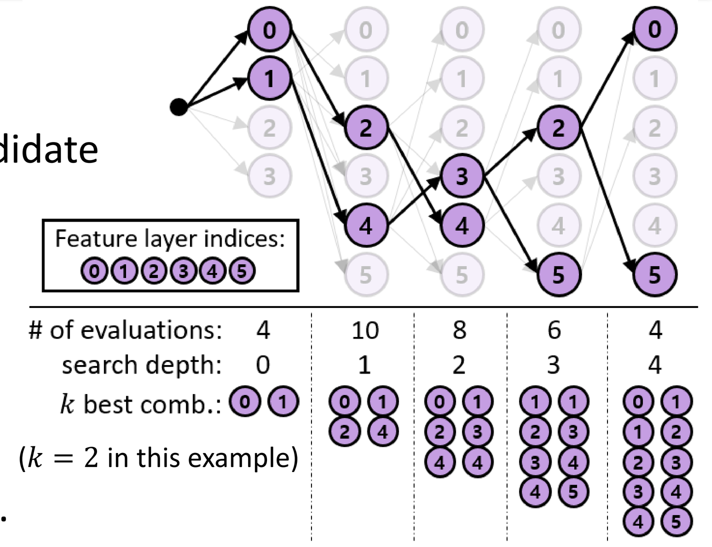


1. Hyperpixel construction:

- Extract L intermediate features maps of CNN pretrained on classification (*e.g.*,ImageNet).
- Take feature maps from the set of layers optimized for correspondence.
  (These layers are pre-selected offline by beam search using small validation data. See below.)
- Concatenate them along channels with upsampling to the size of base map.

Beam search for hyperpixel layers: a breath-first search with a limited memory k

0) Start beam search from the first depth (base) with an empty memory.
1) At the current depth, evaluate the effect of each candidate layer by adding it to all combinations in the memory.
2) Update the memory with k best performing combinations and move on to the next depth.
3) Repeat 1) and 2) until the current search depth reaches predefined maximum search depth.
4) Choose the best combination found along the search.



2. Regularized Hough matching:

- A variant of probabilistic of Hough matching, algorithm of Cho *et al.*'2015
- Reweight appearance similarity by Hough voting to enforce geometric consistency.

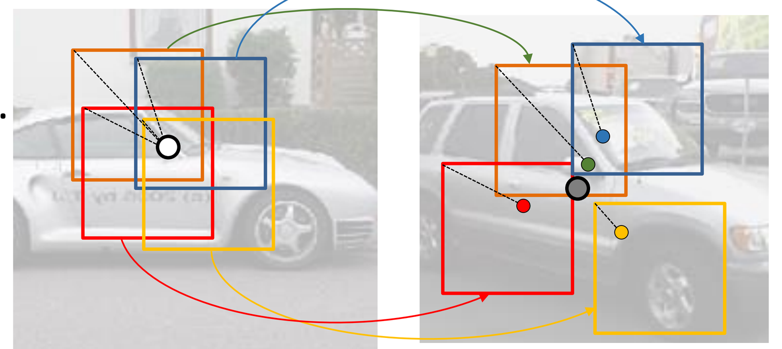
$$p(m_a) = \mathrm{ReLU}\left(\frac{\mathbf{f}\cdot\mathbf{f}'}{\|\mathbf{f}\|\,\|\mathbf{f}'\|}\right)^{d} \qquad p(m|\mathcal{D}) \propto p(m_a)\sum_{\mathbf{x}\in\mathcal{X}} p(m_g|\mathbf{x}) \sum_{m\in\mathcal{H}\times\mathcal{H}'} p(m_a)p(m_g|\mathbf{x})$$

- Regular geometry of hyperpixel enables real-time matching.

3. Flow formation & keypoint transfer:

- Assign a match by nearest-neighbor assignment.
- Evaluate each pair using PCK:

$$\mathrm{PCK}(\mathcal{I},\mathcal{I}') = \frac{1}{M}\sum_{m=1}^{M}\mathbb{1}[\|\mathbf{p}_m - \mathbf{p}'_m\| \le \alpha_\tau \max(w_\tau, h_\tau)]$$

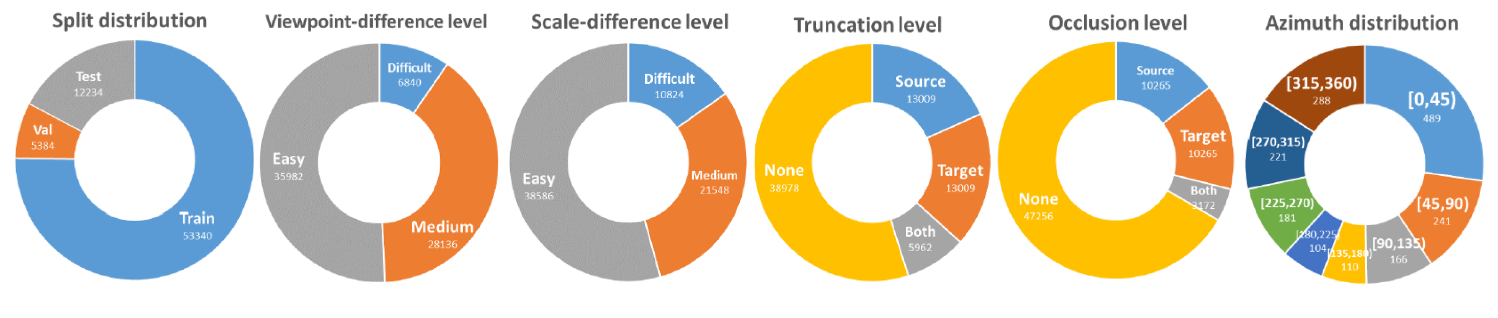


SPair-71k large-scale dataset

- **Comparison between SPair-71k and existing datasets:**

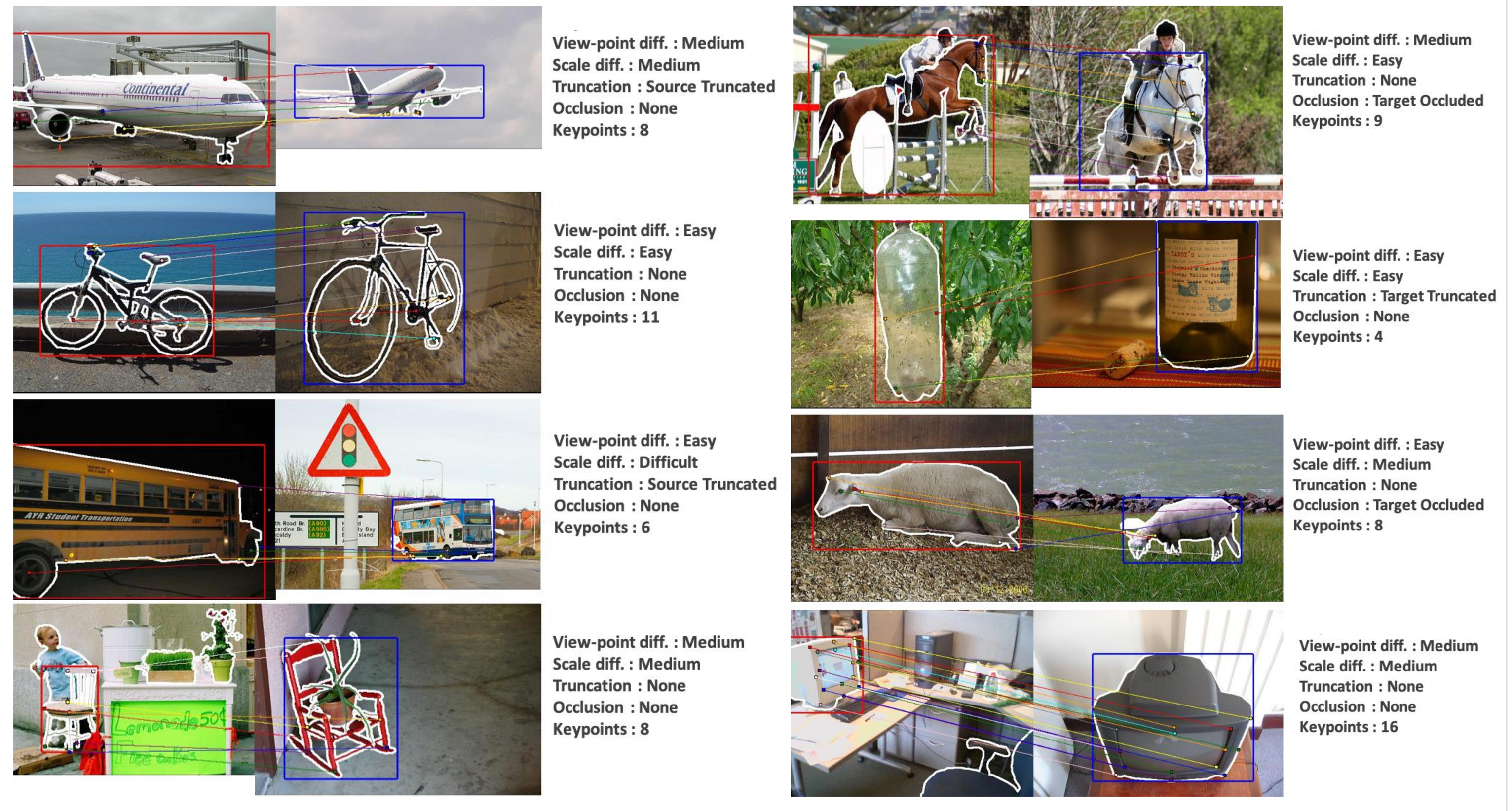
Dataset name		Size (pairs)	Class	Annotations	Characteristics
Caltech-101	Kim *et al.* CVPR'13	1,515	101	object segmentation	tightly cropped images of objects, little background
PASCAL-PARTS	Zhou *et al.* CVPR'15	3,884	20	keypoints (0∼12), azimuth, elevation, cyclo-rotation, body part segmentation	tightly cropped images of objects, little background, part and 3D infomation
Animal-parts	Novotny *et al.* BMVC'16	≈7,000	100	keypoints (1∼6)	keypoints limited to eyes and feet of animals
TSS	Taniai *et al.* CVPR'16	400	9	object segmentation, flow vectors	cropped images of objects, moderate background
PF-WILLOW	Ham *et al.* CVPR'16	900	5	keypoints (10)	center-aligned images, pairs with the same viewpoint
PF-PASCAL	Ham *et al.* TPAMI'18	1,300	20	keypoints (4∼17), bbox.	center-aligned images, pairs with the same viewpoint
SPair-71k (ours)		70,958	18	keypoints (3∼30), azimuth, view-point diff., scale diff., trunc. diff., occl. diff., object seg., bbox.	large-scale data with diverse variations, rich annotations, clear dataset splits

- **Dataset statistics:**



Type	View-point diff.			Scale diff.			Truncation diff.				Occlusion diff.			
	easy	medi	hard	easy	medi	hard	none	src	tgt	both	none	src	tgt	both
Train	26,466	21,646	5,228	29,248	16,184	7,908	29,184	9,796	9,796	4,564	35,330	7,737	7,737	2,536
Val	2,862	2,016	506	2,880	1,570	934	2,744	1,047	1,047	546	3,760	722	722	180
Test	6,654	4,474	1,106	6,458	3,794	1,982	7,050	2,166	2,166	852	8,166	1,806	1,806	456
All	35,982	28,136	6,840	38,586	21,548	10,824	38,978	13,009	13,009	5,962	47,256	10,265	10,265	3,172

- **Example pairs and annotations:**



Experimental results

- **Performance on standard benchmarks of semantic correspondence:**

Methods		Supervision	PF-PAS. (@α_{img}) 0.05	PF-PAS. 0.1	PF-WIL. (@α_{bbox}) 0.05	PF-WIL. 0.1	Caltech-101 LT-ACC	Caltech-101 IoU	time (ms)
PF_{HOG}	Ham *et al.* CVPR'16	-	31.4	62.5	28.4	56.8	0.78	0.50	> 1000
$CNNGeo_{res101}$	Rocco *et al.* CVPR'17	synthetic warp (self-supervised)	41.0	69.5	36.9	69.2	0.79	0.56	40
$A2Net_{res101}$	Seo *et al.* ECCV'18		42.8	70.8	36.3	68.8	0.80	0.57	53
$DCTM_{CAT\text{-}FCSS}$	Kim *et al.* ICCV'17	image labels (weakly-supervised)	34.2	69.6	38.1	61.0	0.83	0.52	-
$Weakalign_{res101}$	Rocco *et al.* CVPR'18		49.0	74.8	37.0	70.2	0.85	0.63	41
$NC\text{-}Net_{res101}$	Rocco *et al.* NeurIPS'18		54.3	78.9	33.8	67.0	0.85	0.60	261
$RTNs_{res101}$	Kim *et al.* NeurIPS'18		55.2	75.9	41.3	71.9	-	-	376
$UCN_{GoogLeNet}$	Choy *et al.* NeurIPS'16	keypoints	29.9	55.6	24.1	54.0	-	-	-
$SCNet_{vgg16}$	Han *et al.* ICCV'17		36.2	72.2	38.6	70.4	0.79	0.51	> 1000
$NN\text{-}Cyc_{res101}$	Laskar *et al.* WACV'19		55.1	85.7	40.5	72.5	0.86	0.62	-
HPF_{res50} (ours)		keypoints (validation only)	60.5	83.4	46.5	72.4	**0.88**	**0.64**	34 (**19**)
HPF_{res101} (ours)			60.1	84.8	45.9	74.4	0.87	0.63	63
$HPF_{res101\text{-}FCN}$ (ours)			**63.5**	**88.3**	**48.6**	**76.3**	0.87	0.63	-
HPF_{res101} (k=1)		keypoints (validation only, small set)	$59.4_{\pm0.89}$	$83.9_{\pm1.14}$	$44.5_{\pm0.90}$	$72.5_{\pm1.22}$	0.87	0.63	-
HPF_{res101} (k=2)			$58.3_{\pm1.33}$	$84.5_{\pm0.77}$	$44.7_{\pm0.92}$	$73.1_{\pm1.05}$	0.87	0.63	-
HPF_{res101} (k=3)			$59.4_{\pm1.16}$	$84.5_{\pm0.27}$	$45.1_{\pm0.55}$	$73.4_{\pm0.52}$	0.87	0.63	-
HPF_{res101} (random)		-	$44.5_{\pm11.11}$	$74.7_{\pm6.46}$	$32.8_{\pm8.12}$	$62.4_{\pm6.67}$	0.85	0.55	-

Small set exp.: layer search using **ONLY** k random pairs per class ($20 * k$ pairs total)
Results with little supervisory signal (**20 pairs**) is comparable as using all data (**308 pairs**).

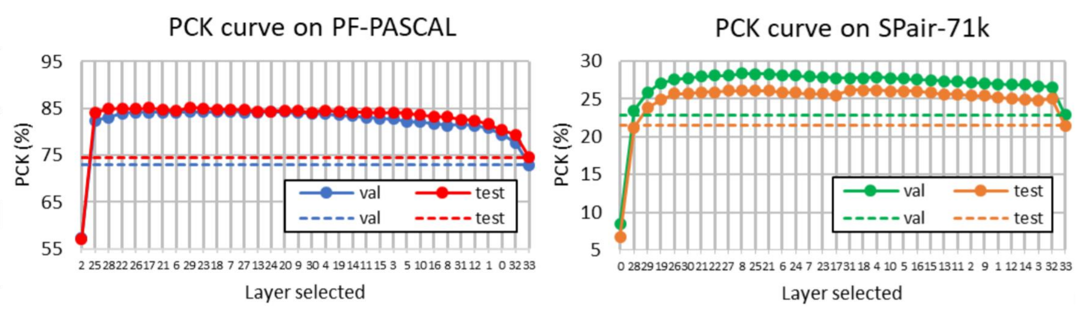
- **Performance on SPair-71k dataset:**

Methods		aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	dog	horse	moto	person	plant	sheep	train	tv	all
Transferred models	$CNNGeo_{res101}$	21.3	15.1	34.6	12.8	31.2	26.3	24.0	30.6	11.6	24.3	20.4	12.2	19.7	15.6	14.3	9.6	28.5	28.8	18.1
	$A2Net_{res101}$	20.8	17.1	37.4	13.9	33.6	29.4	26.5	34.9	12.0	26.5	22.5	13.3	21.3	20.0	16.9	11.5	28.9	31.6	20.1
	$WeakAlign_{res101}$	23.4	17.0	41.6	14.6	37.6	28.1	26.6	32.6	12.6	27.9	23.0	13.6	21.3	22.2	17.9	10.9	31.5	34.8	21.1
	$NC\text{-}Net_{res101}$	24.0	16.0	45.0	13.7	35.7	25.9	19.0	50.4	14.3	32.6	27.4	19.2	21.7	20.3	20.4	13.6	**33.6**	**40.4**	26.4
SPair-71k trained models	$CNNGeo_{res101}$	23.4	16.7	40.2	14.3	36.4	27.7	26.0	32.7	12.7	27.4	22.8	13.7	20.9	21.0	17.5	10.2	30.8	34.1	20.6
	$A2Net_{res101}$	22.6	18.5	42.0	**16.4**	37.9	**30.8**	26.5	35.6	13.3	29.6	24.3	16.0	21.6	22.8	20.5	13.5	31.4	36.5	22.3
	$WeakAlign_{res101}$	22.2	17.6	41.9	15.1	**38.1**	27.4	**27.2**	31.8	12.8	26.8	22.6	14.2	20.0	22.2	17.9	10.4	32.2	35.1	20.9
	$NC\text{-}Net_{res101}$	17.9	12.2	32.1	11.7	29.0	19.9	16.1	39.2	9.9	23.9	18.8	15.7	17.4	15.9	14.8	9.6	24.2	31.1	20.1
HPF_{res50} (ours)		**25.3**	18.5	47.6	14.6	37.0	22.9	18.3	51.1	16.7	31.5	30.8	19.1	23.7	23.8	**23.5**	14.4	30.8	37.2	27.2
HPF_{res101} (ours)		25.2	**18.9**	**52.1**	15.7	38.0	22.8	19.1	**52.9**	**17.9**	**33.0**	**32.8**	**20.6**	**24.4**	**27.9**	21.1	**15.9**	31.5	35.6	**28.2**

SOTA on new benchmark SPair-71k that has pairs with large view-point and scale differences.

- **Ablation study:**

Matching module	PF-PASCAL $\alpha_{img} = 0.1$	PF-WILLOW $\alpha_{bbox} = 0.1$
NN w/ ($d = 1$)	69.0	60.9
RHM w/ ($d = 1$)	81.4	68.6
RHM w/ ($d = 2$)	84.4	73.3
RHM w/ ($d = 3$)*	**84.8**	**74.4**
RHM w/ ($d = 4$)	84.8	74.1
RHM w/ ($d = 5$)	84.5	73.9



Only **a few layers are sufficient** to achieve a comparable performance with the best one.

- **Qualitative results:**

(a) source image (b) target image (c) CNNGeo (d) A2Net (e) WeakAlign (f) NC-Net (g) HPF (ours)