CNN-based Patch Matching for Optical Flow with Thresholded Hinge Embedding Loss

Christian Bailer, Kiran Varanasi, Didier Stricker
German Research Center for Artificial Intelligence

Overview

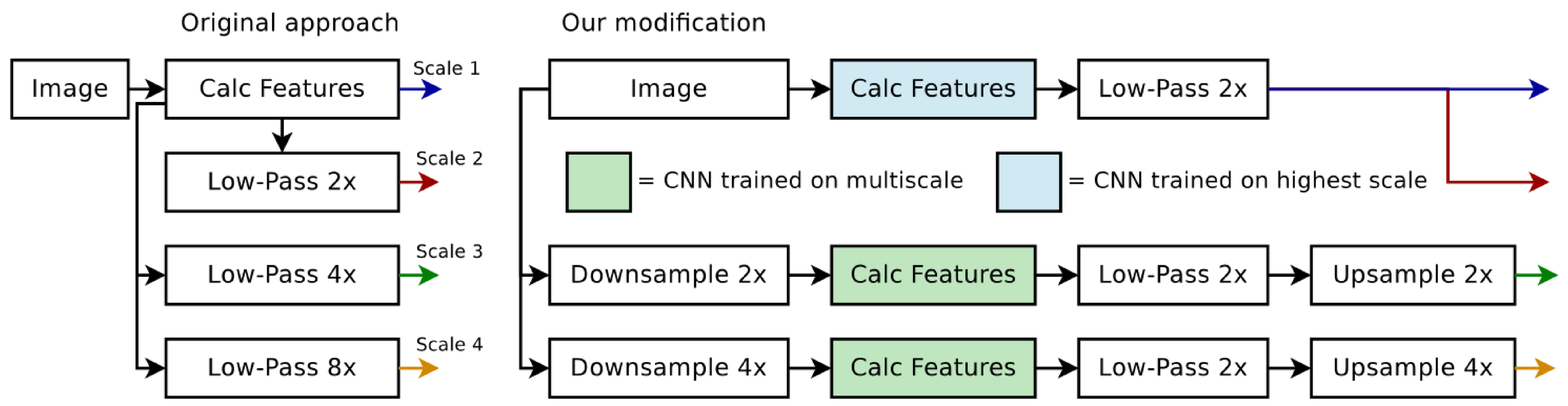
We propose a new CNN-based Patch matching approach for optical flow. Many of our contributions are not limited to optical flow but interesting for general feature matching.

Our main contributions are:

- A new loss function for feature and patch matching based on Siamese networks. It outperforms existing losses and allows faster training.
- New evaluation measure for feature and patch matching.
- We show: low-pass filtering of feature maps created by CNNs improves matching robustness.
- Multi-scale CNN based feature creation approach tailored for optical flow.
- We obtain state-of-the art results on on KITTI and MPI-Sintel.

Our multi-scale pipeline

Our matching pipeline is similar to the multi-scale patch matching pipeline of FlowFields [1]. However, we introduce some modifications advantageous for CNNs (see figure below):



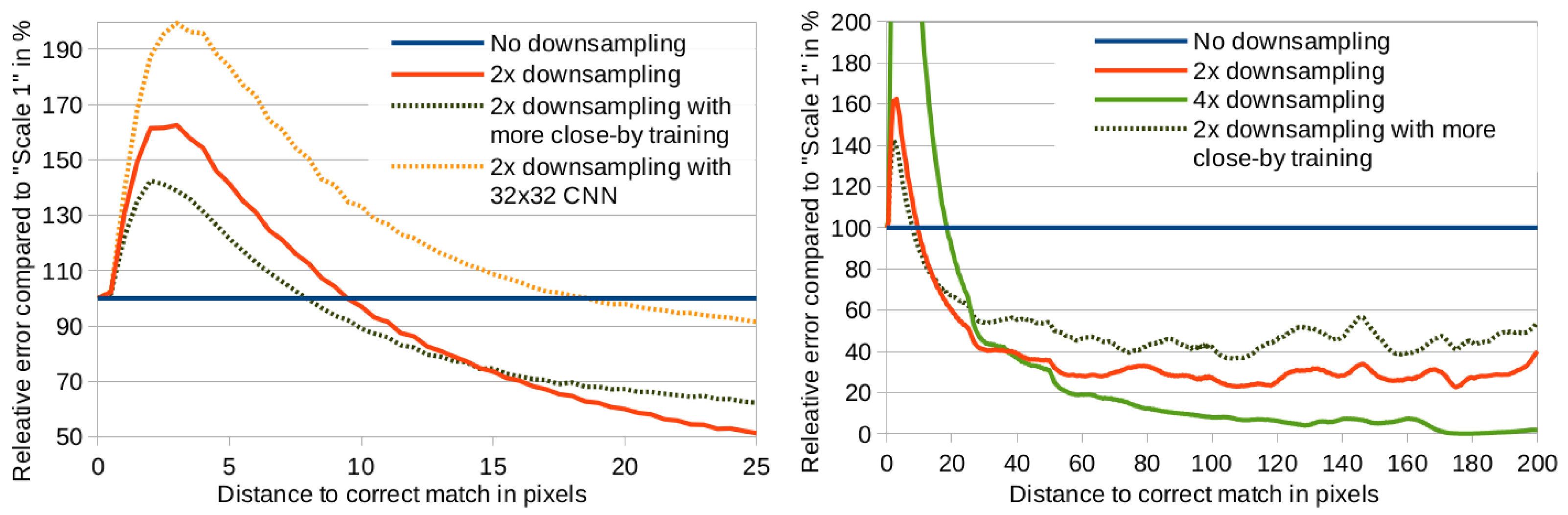
Main differences to [1] are:

- We calculate features between up- and downsampling instead of before up+downsampling (=low-pass).
- We use two different CNNs to calculate features (one trained on all scales, one only on highest scale).
- We use low-pass to filter after feature calculation to increase matching invariance.
- We omit one scale as CNNs are robust enough to not require it (and even perform worth when used)

Experiments:

Method	EPE	EPE >3 px. noc.
Our pipeline	**2.60 px**	**4.95%**
Our pipeline, no "low-pass 2x" filter	2.80 px	5.21%
Our pipeline, not omitting one scale	2.98 px	5.66%
FlowFields pipeline [1] with our CNNs	3.08 px	5.48%

Performance of different pipelines



Analysis of robustness of CNN based matching on downsampled images.
More close-by training = CNN trains more negative matches very close to positive match

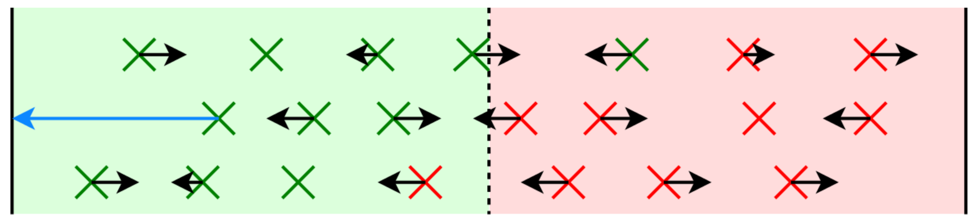
Feature learning

We train our network with a Siamese architecture. In contrast to related work we use a novel thresholded loss that does not unnecessarily minimize matching errors for correct matches.
For training details see paper.

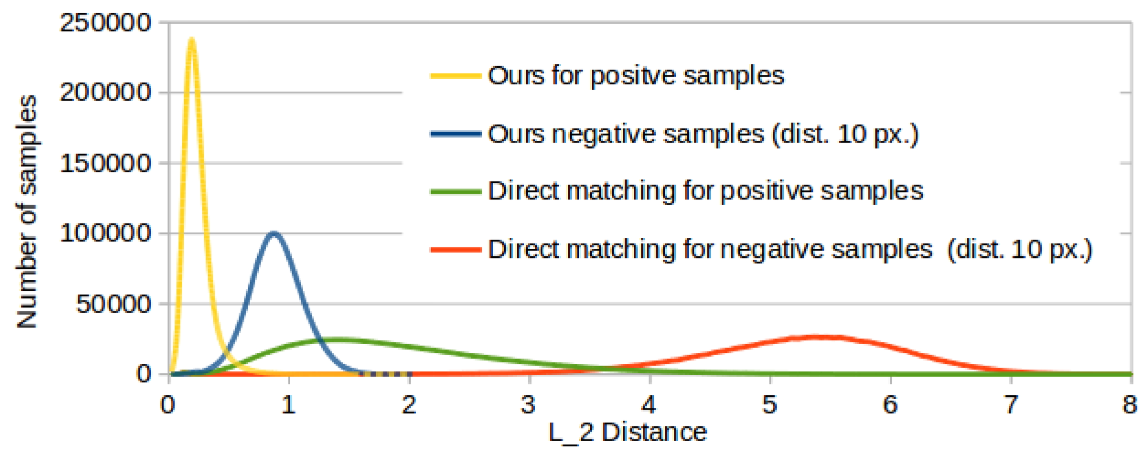
Loss Function:

	Common/Hinge Loss	Direct match loss	Our loss
Patches p_1 and p_2^+ match	$L_2(p_1,p_2^+) \rightarrow 0$	$L_2(p_1,p_2^+) < L_2(p_1,p_2^-)$	$L_2(p_1,p_2^+) \rightarrow < t$
p_1 and p_2^- do not match	$L_2(p_1,p_2^-) \rightarrow > m$		$L_2(p_1,p_2^-) \rightarrow > m\text{-}t$
EPE > 3px. (experiment)	7.26%	5.89%	**4.95%**

$L_2(x,y) = \|D(x) - D(y)\|_2$. Table shows what losses aim for. If fulfilled loss is zero. Our loss is especially often zero (>80%). As we do not backpropagate zero losses this makes our loss also very fast in training.



Common losses push the matching error of matching patches to zero (blue arrow in illustration below). This comes at high cost (black arrows) without any advantage (the pushed point is already fine).



Direct matching and **our loss** focus on problematic patches, but ours has a much smaller variance (with similar average gap: $L_2(p_1,p_2^+)$ - $L_2(p_1,p_2^-)$). Small gap + large variance = unreliable in test data.

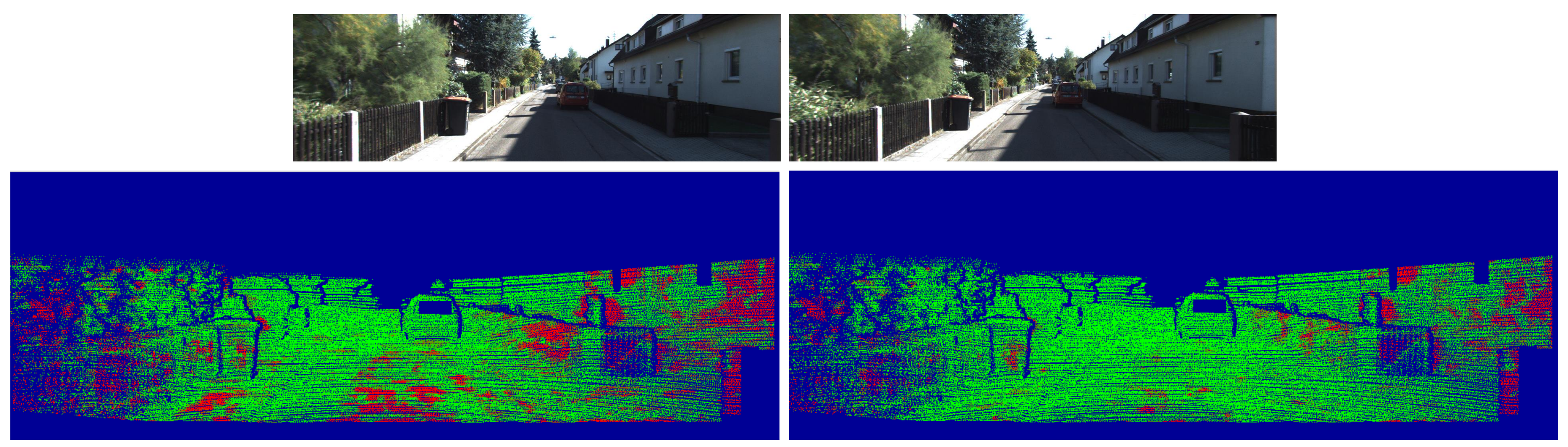
Low-pass filtering feature maps:

Low-pass filtering increases invariance (but also ambiguity). To some extend it increases robustness as CNNs do not to learn this effect completely. We hope it can be applied on other applications (like R-CNN).

Evaluation measure:

Matching is not a classification problem but a binary decision problem → do not use ROC or PR but the probability that the correct patch p_2^+ matches p_1 better than an arbitrary wrong one p_2^-:

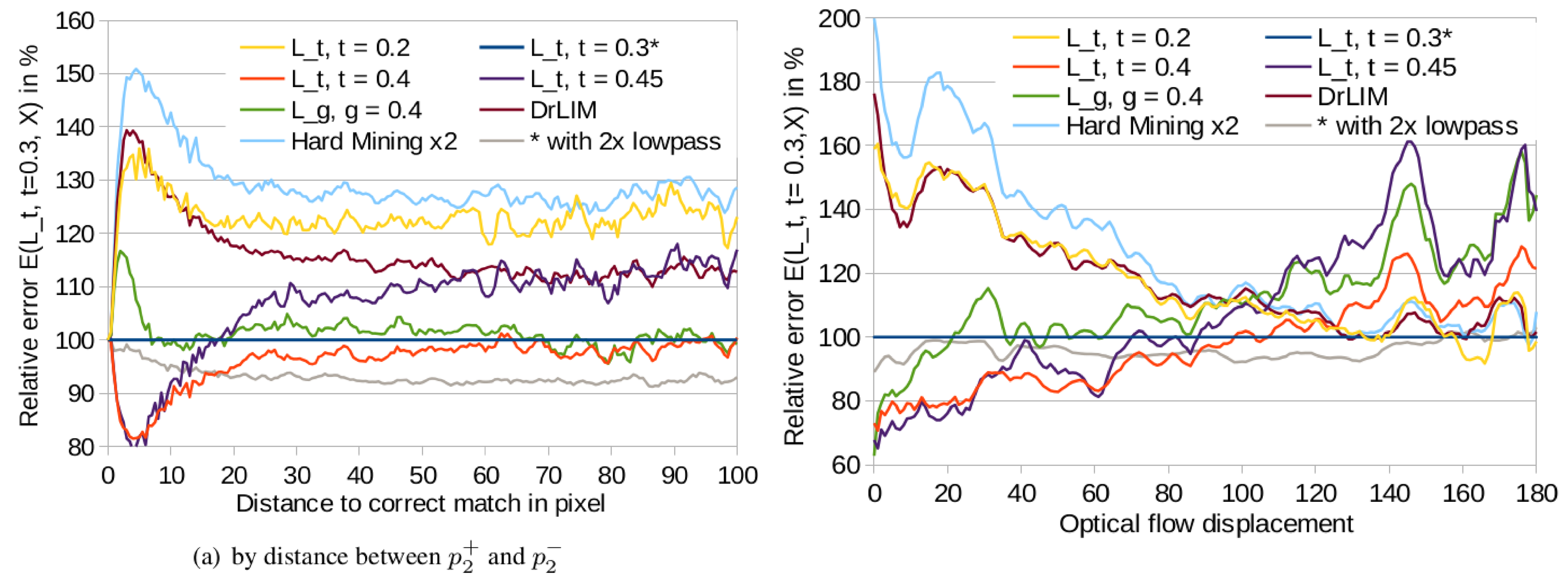
$$r = \sum_{p_1} \frac{P(L_2(p_1,p_2^+) < L_2(p_1,p_2^-))}{|p_1|}$$



hinge loss **our loss**

Pixelwise matching robustness r for our loss and the common hinge loss.
Red: r is small, Green: r is large, Blue: No ground truth.

Experiment on loss function behavior



We visualize the matching robustness r (see middle row in poster)for different distances (between correct p_2^+ and false match p_2^-) and different flow displacements.
L_t = our loss, L_h = Hinge loss, L_g = Direct match loss, others = see paper
For better visualization we plot not r but r/r0 while r0 is for L_t , t = 0.3

Public Results

KITTI 2012:

Method	EPE	EPE >3 px. noc.	Runtime all	Runtime CNN
Ours (56x56)	**3.0 px**	**4.89%**	**23s**	4.5s (**2s***)
Patchbatch(71x71)	3.3 px	4.92%	60s	27.5s
Patchbatch(51x51)	3.3 px	5.29%	50	37.5s
FlowFields	3.5 px	5.77%	23	-
Patchbatch FAST	-	5.94%	25.5s	2.5s

*Fast approach not tested on test set but comparable to slow approach on training set.

KITTI 2015:

Method	EPE >3 px. noc. background	EPE >3 px. noc. foreground	EPE >3 px.	Runtime
Ours	**8.91%**	**20.78%**	**19.44%**	**23s**
Patchbatch	10.06%	26.21%	21.69%	50s
DeepDiscreteFlow	10.44%	25.86%	21.92%	60s

MPI-Sintel (Final):

Method	EPE	EPE noc.
Ours	**5.363**	**2.303**
DeepDiscreteFlow	5.728	2.623
FlowFields	5.810	2.621
CPM-Flow	5.960	2.990
PatchBatch	6.783	3.507

For fast feature calculation on whole images we use the approach detailed and generalized in our follow-up paper [2] (A brief version is also in our supplementary material).

References

[1] Bailer Christian, Bertram Taetz, and Didier Stricker. "Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation. ICCV 2015.
[2] Bailer Christian, Tewodros Habtegebrial, Kiran Varanasi and Didier Stricker. Fast Dense Feature Extraction with Convolutional Neural Networks that have Pooling or Striding Layers. BMVC 2017