

Efficient Sparse-to-Dense Optical Flow Estimation using a Learned Basis and Layers

Jonas Wulff, Michael J. Black

Max Planck Institute for Intelligent Systems, Tübingen, Germany.

Despite decades of research, a fast optical flow algorithm, that is also accurate, remains an elusive goal. While recent optical flow methods such as [6] are highly accurate, for many applications speed is often as important. Most current methods take several seconds (or minutes) per frame, making them impractical for long sequences. The fastest optical flow algorithms run in under a second, but rely on GPUs [7], which are not available on all platforms. We address both accuracy and speed and propose a new approach called PCA-Flow that is based on sparse feature matches followed by interpolation. Both steps are fast to compute. To increase accuracy, especially at motion boundaries, we extend PCA-Flow to a layered framework. Both approaches are significantly faster than methods with comparable accuracies.

PCA-Flow. The basic assumption for our work is that an arbitrary optical flow field can be approximated as a weighted sum of a small number of basis flow fields,

$$\tilde{\mathbf{u}} = \sum_{n=1}^N w_n \mathbf{b}_n. \quad (1)$$

Previously such linear flow models were only used in small image patches [2] or simple, inherently low-dimensional scenarios such as driving vehicles [4]. In contrast, we use a linear model for full-frame optical flow in arbitrary scenes. To find a suitable basis, we first compute optical flow for four Hollywood movies, resulting in 8 hours of flow data. We then compute the basis that spans this optical flow using a robust PCA method [3], and take the first 250 eigenvectors in both horizontal and vertical directions as the basis vectors \mathbf{b}_n . Figure 1(a) shows an example.

A linear model as in (1) has three main advantages. First, given a fixed basis, the optical flow field is completely determined by the weights $\mathbf{w} = (w_1, \dots, w_N)^\top$, yielding a much more compact representation than the 2D vector field. Second, a domain-specific prior (for example for automotive scenarios) on \mathbf{w} can be learned even from small training sets. Third, given sparse, matched features and taking the prior into account, we can efficiently compute \mathbf{w} as the solution to a robustified, Tikhonov-regularized least squares problem.

Figure 2(d) shows an example for an optical flow field computed by PCA-Flow. Due to the low dimensionality of the optical flow basis, motion boundaries are blurry. However, in terms of average endpoint error our method beats the widely used methods Classic+NL and LDOF, while requiring on average 190 ms per flow field on a CPU. Figure 1(b) shows PCA-Flow and other state of the art algorithms in the accuracy/runtime plane on KITTI. While many algorithms are more accurate than ours, they are all several times slower.

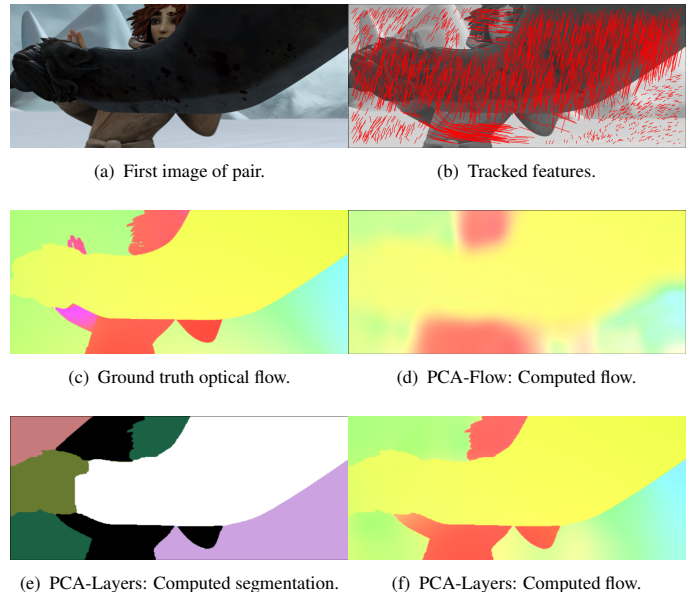


Figure 2: Results of PCA-Flow and PCA-Layers.

expensive [5]. Here, we cluster matched features into layers using Expectation Maximization (EM), and use PCA-Flow to efficiently compute the motion of each layer. The computation of a dense flow field then becomes a labelling problem, in which each pixel is assigned to one of the motion layers. We formulate this as a Markov Random Field, taking into account brightness constancy, color consistency and spatial compactness within the layers, and a smoothness term that incorporates image content. This smoothness term leads to good extrapolation of flow values to image regions that are not well covered by matched features.

Figure 2(e) shows an example for the computed layer segmentation, and Fig. 2(f) shows the resulting optical flow. Compared to Fig. 2(d), the motion boundaries are much better preserved. PCA-Layers takes on average 3.2 seconds per pair of frames, and in MPI-Sintel ranks in place 10 of 36 in the final pass and in place 9 of 36 in the clean pass, beating popular methods such as MDP-Flow. The next fastest method with similar accuracy, SparseFlow, is over 2.5 times slower, taking 10 s per frame.

More details on the optimization and more results can be found in the paper. The code for PCA-Flow, PCA-Layers, and the learned optical flow basis is available at [1].

[1] <http://pcaflow.is.tue.mpg.de>.

[2] D.J. Fleet, M.J. Black, Y. Yacoob, and A.D. Jepson. Design and use of linear models for image motion analysis. *IJCV*, 36(3):171–193, 2000.

[3] S. Hauberg, A. Feragen, and M.J. Black. Grassmann averages for scalable robust PCA. *CVPR*, pp. 3810–3817, 2014.

[4] R. Roberts, C. Potthast, and F. Dellaert. Learning general optical flow subspaces for egomotion estimation and detection of motion anomalies. *CVPR*, pp. 57–64, 2009.

[5] D. Sun, E.B. Sudderth, and M.J. Black. Layered segmentation and optical flow estimation over time. *CVPR*, pp. 1768–1775, 2012.

[6] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. DeepFlow: Large displacement optical flow with deep matching. *ICCV*, pp. 1385–1392, 2013.

[7] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime TV-L1 optical flow. *Pattern Recog.*, Springer LNCS 4713, pp. 214–223, 2007.

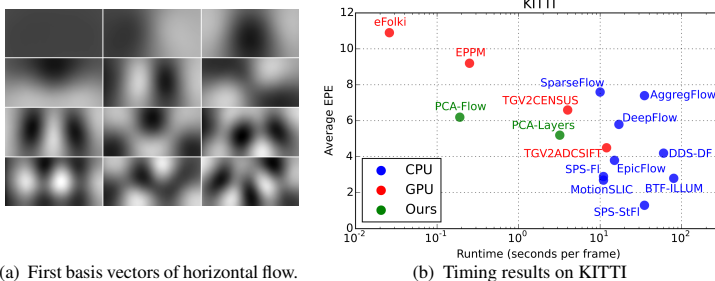


Figure 1: Learned flow basis and timing results

PCA-Layers. To increase accuracy at the motion boundaries, we extend PCA-Flow to a layered model, PCA-Layers. This assumes that an optical flow field can be decomposed into segments, each containing very simple motion. Unfortunately so far, layered models have been computationally