

Learning Human Optical Flow

Anurag Ranjan¹
aranjan@tuebingen.mpg.de

Javier Romero^{*,2}
javier@amazon.com

Michael J. Black¹
black@tuebingen.mpg.de

¹ MPI for Intelligent Systems
Tübingen, Germany

² Amazon Inc.

Abstract

The optical flow of humans is well known to be useful for the analysis of human action. Given this, we devise an optical flow algorithm specifically for human motion and show that it is superior to generic flow methods. Designing a method by hand is impractical, so we develop a new training database of image sequences with ground truth optical flow. For this we use a 3D model of the human body and motion capture data to synthesize realistic flow fields. We then train a convolutional neural network to estimate human flow fields from pairs of images. Since many applications in human motion analysis depend on speed, and we anticipate mobile applications, we base our method on SpyNet with several modifications. We demonstrate that our trained network is more accurate than a wide range of top methods on held-out test data and that it generalizes well to real image sequences. When combined with a person detector/tracker, the approach provides a full solution to the problem of 2D human flow estimation. Both the code and the dataset are available for research.

1 Introduction

A significant fraction of videos on the Internet contain people moving [18] and the literature suggests that optical flow plays an important role in understanding human action [25, 39]. Several action recognition datasets [28, 39] contain human motion as a major component. The 2D motion of humans in video, or *human flow*, is an important feature that provides a building block for systems that can understand and interact with humans. Human flow is useful for various applications including analyzing pedestrians in road sequences, motion-controlled gaming, activity recognition, human pose estimation system, etc.

Despite this, optical flow has previously been treated as a generic, low-level, vision problem. Given the importance of people, and the value of optical flow in understanding them, we develop a flow algorithm that is specifically tailored to humans and their motion. Such motions are non-trivial since humans are complex, articulated, objects that vary in shape, size and appearance. They move quickly, self occlude, and adopt a wide range of poses.

Our goal is to obtain more accurate 2D motion estimates for human bodies by training a flow algorithm specifically for human movement. To do so, we create a large and realistic dataset of humans moving in virtual worlds with ground truth optical flow (Fig. 1(a)). We

*This work was done by JR while at MPI.



Figure 1: (a) We simulate human motion in virtual world creating an extensive dataset with images (top row) and flow fields (bottom row); color coding from [1]. (b) We train an existing deep network for human motion estimation and show that it performs better when trained on our dataset and (c) generalizes to human motions in real world scenes.

train a neural network based on SPyNet [53] using this dataset and show that it outperforms state of the art optical flow on the test sequences of this dataset (Fig. 1(b)). Furthermore we show that it generalizes to real video sequences (Fig. 1(c)). Here we also extend SPyNet, making it end-to-end trainable.

Several datasets and benchmarks [4, 5, 17] have been established to drive the progress in optical flow. We argue that these datasets are insufficient for the task of human motion estimation and, despite its importance no attention has been paid to datasets and algorithms for human flow. One of the main reasons is that dense human motion is extremely difficult to capture accurately in real scenes. Without ground truth, there has been little work focused specifically on estimating human optical flow. To advance research on this problem, the community needs a dataset tailored to human flow.

A key observation is that recent work has shown that optical flow methods trained on synthetic data [9, 22, 53] generalize relatively well to real data. Additionally, these methods obtain state of the art results with increased realism of the training data [15, 53]. This motivates our effort to create a dataset designed for human motion.

To that end, we use the SMPL body model [30] to generate about a hundred thousand different human shapes. We then place them on random indoor backgrounds and simulate human activities like running, walking, dancing etc. using motion capture data [29]. Thus, we create a large virtual dataset that captures the statistics of natural human motion. We then train a deep neural network based on spatial pyramids [53] and evaluate its performance for estimating human motion. While the dataset can be used to train any flow method, we choose SpyNet because it is compact and computationally efficient.

In summary, our major contributions are: 1) we provide the “Human Flow dataset” with 146,020 frame pairs of human bodies in motion with realistic textures and backgrounds; 2) we show that our network outperforms previous optical flow methods by 30% on the Human Flow dataset, and it generalizes to real world scenes; 3) we extend SPyNet to be fully end-to-end trainable; 4) our neural network is very small (7.8 MB for the network parameters) and runs in real time (32fps), hence it can be potentially used for embedded applications; 5) we provide data, code, and the trained model¹ for research purposes.

¹<http://github.com/anuragranj/humanflow>

2 Related Work

Human Motion. Human motion can be understood from 2D motion. Early work focused on the movement of 2D joint locations [26] or simple motion history images [8]. Optical flow is also a useful cue. Black et al. [9] use principal component analysis (PCA) to parametrize human motion but use noisy flow computed from image sequences for training data. More similar to us, Fablet and Black [10] use a 3D articulated body model and motion capture data to project 3D body motion into 2D optical flow. They then learn a view-based PCA model of the flow fields. We use a more realistic body model to generate a large dataset and use this to train a CNN to directly estimate dense human flow from images.

Only a few works in pose estimation have exploited human motion and, in particular several methods [13, 14] use optical flow constraints to improve 2D human pose estimation in videos. Similar work [6, 34] propagates pose results temporally using optical flow to encourage time consistency of the estimated bodies. Apart from its application in warping between frames, the structural information existing in optical flow has been used for pose estimation alone [38] or in conjunction with an image stream [11].

Learning Optical Flow. There is a long history of optical flow estimation, which we do not review here. Instead, we focus on the relatively recent literature on learning flow. Early work looked at learning flow using Markov Random Fields [14], PCA [12], or shallow convolutional models [7]. Other methods also combine learning with traditional approaches, formulating flow as a discrete [20] or continuous [36] optimization problem.

The most recent methods employ large datasets to estimate optical flow using deep neural networks. Voxel2Voxel [10] is based on volumetric convolutions to predict optical flow using 16 frames simultaneously but does not perform well on benchmarks. Other methods [9, 22, 35] compute two frame optical flow using an end-to-end deep learning approach. FlowNet [9] uses the Flying Chairs dataset [9] to compute optical flow in an end to end deep network. FlowNet 2.0 [22] uses stacks of networks from FlowNet and performs significantly better, particularly for small motions. Ranjan and Black [35] propose a Spatial Pyramid Network that employs a small neural network on each level of an image pyramid to compute optical flow. Their method uses a much smaller number of parameters and achieves similar performance as FlowNet [9] using the same training data. Since the above methods are not trained with human motions, they do not perform well on our Human Flow dataset.

Optical Flow Datasets. Several datasets have been developed to facilitate training and benchmarking of optical flow methods. Middlebury is limited to small motions [7], KITTI is focused on rigid scenes and automotive motions [17], while Sintel has a limited number of synthetic scenes [6]. These datasets are mainly used for evaluation of optical flow methods and are generally too small to support training neural networks.

To learn optical flow using neural networks, more datasets have emerged that contain examples on the order of tens of thousands of frames. The Flying Chairs [9] dataset contains about 22,000 samples of chairs moving against random backgrounds. Although it is not very realistic or diverse, it provides training data for neural networks [9, 35] that achieve reasonable results on optical flow benchmarks. Even more recent datasets [15, 13] for optical flow are especially designed for training deep neural networks. Flying Things [13] contains tens of thousands of samples of random 3D objects in motion. The Monkaa and Driving scene datasets [13] contain frames from animated scenes and virtual driving respectively. Virtual KITTI [15] uses graphics to generate scenes like those in KITTI and is two orders of magnitude larger. Recent synthetic datasets [16] show that synthetic data can train networks that generalize to real scenes.

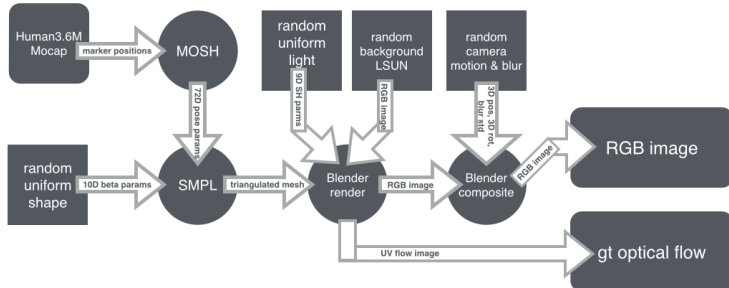


Figure 2: Pipeline for generating the RGB frames and ground truth optical flow for the Human Flow dataset.

For human bodies, the SURREAL dataset [40] uses 3D human meshes rendered on top of images to train networks for 2D pose estimation, depth estimation, and body part segmentation. While not fully realistic, they show that this data is sufficient to train methods that generalize to real data. We go beyond their work to address the problem of optical flow.

3 The Human Flow Dataset

Our approach generates a realistic dataset of synthetic human motions by simulating them against different realistic backgrounds. As shown in Figure 2, we use the SMPL model [30] to generate a wide variety of different human shapes and appearances. We use Blender² as a rendering engine to generate synthetic image frames and optical flow. In the rest of the section, we describe each component of our pipeline shown in Figure 2.

Body model and Rendering Engine. The main challenge in the generation of realistic human optical flow is modeling realistic articulated motions. As shown in Figure 2, we use the SMPL model [30], parameterized by pose and shape parameters to change the body posture and identity. The model also contains a UV appearance map that allows us to change the skin tone, face features and clothing texture of the model. A key component of Blender in this project is its *Vector pass*. This render pass is typically used for producing motion blur, and it produces the motion in image space of every pixel; i.e. ground truth optical flow. We are mainly interested in the result of this pass, together with the color rendering of the textured bodies.

Body Poses. In order to obtain a varied set of poses, we use motions from the Human3.6M dataset [23]. Human3.6M contains five subjects for training (S1, S5, S6, S7, S8) and two for testing (S9, S11). Each subject performs 15 actions twice, resulting in 1,559,985 frames for training and 550,727 for testing. These sequences are subsampled at a rate of $16\times$, resulting in 97,499 training and 34,420 testing poses from Human3.6M. The pose data is then converted into SMPL body models using MoSh [29]. We limit each of our pose sequences to 20 frames.

Body shapes. To maximize the variety of the data, each sequence of 20 frames uses a random body shape drawn from a uniform distribution of SMPL shape parameters bounded by $[-3, 3]$ standard deviations for each shape coefficient according to the shape distribution in CAESAR [57]. Using a parametric distribution ensures that each sub-sequence of frames has a unique body shape. A uniform distribution has more extreme shapes than the Gaussian

²<https://www.blender.org/>

distribution inferred originally from CAESAR, while avoiding unlikely shapes by strictly bounding the coefficients.

Scene Illumination. Optical flow estimation should be robust to different scene illumination. In order to achieve this invariance, we illuminate the bodies with Spherical Harmonics lighting [19]. Spherical Harmonics define basis vectors for light directions that are scaled and linearly combined. This compact parameterization is particularly useful for randomizing the scene light. The linear coefficients are randomly sampled with a slight bias towards natural illumination. The coefficients are uniformly sampled between -0.7 and 0.7 , apart from the ambient illumination (which is strictly positive and a minimum of 0.3) and the vertical illumination (which is strictly negative to prevent illumination from below).

Body texture. To provide a varied set of appearances to the bodies in the scene, we use textures from two different sources. A wide variety of human skin tones is extracted from the CAESAR dataset [6]. Given SMPL registrations to CAESAR scans, the original per-vertex color in the CAESAR dataset is transferred into the SMPL texture map. Since fiducial markers were placed on the bodies of CAESAR subjects, we remove them from the textures and inpaint them to produce a natural texture. The main drawback of CAESAR scans is their homogeneity in terms of outfit, since all of the subjects wore grey shorts and the women wore sports bras. In order to increase the clothing variety, we also use textures extracted from 3D scans. A total of 772 textures from 7 different subjects with different clothes were captured. All textures were anonymized by replacing the face by the average face in CAESAR, after correcting it to match the skin tone of the texture. The datasets were partitioned 70%|30% into training and testing, and each texture dataset was sampled with a 50% chance.

Background texture. The other crucial component of image appearance is the background. Since human motion rarely happens in front of clean and easily segmentable scenes, realistic backgrounds should be included in the synthetic scenes. We found that using random indoor images from the LSUN dataset [4] as background provided a good compromise between simplicity and the complex task of generating varied full 3D environments. We used 417,597 images from LSUN categories kitchen, living room, bedroom and dining room. The background images were placed as billboards 9 meters from the camera, and were not affected by the spherical harmonics lighting.

Increasing image realism. One of the main differences between synthetic and real images are the imperfections existing in the latter. Fully sharp images rendered with perfectly static virtual cameras do not represent well the images captured in real situations. In order to increase realism, we introduced three types of images imperfections. First, in 30% of the generated images we introduced camera motion between frames. This motion perturbs the location of the camera with Gaussian noise of 1 centimeter standard deviation, and rotation noise of 0.2 degrees standard deviation per dimension in an Euler angle representation. Second, motion blur was added to the scene. The motion blur was implemented with the *Vector Blur Node* in Blender, and integrated over 2 frames sampled with 64 steps between the beginning and end point of the motion. Finally, general image blur was added to 30% of the images, as Gaussian blur with a standard deviation of 1 pixel.

Dataset Details. In comparison with other optical flow datasets, our dataset is larger by an order of magnitude, containing 135,153 training frames and 10,867 test frames with optical flow ground truth. We keep the resolution small at 256×256 to facilitate easy deployment for training neural networks. We show the comparisons in Table 1(a). This also speeds up the rendering process in Blender for generating large amounts of data. Our data is extensive, containing a wide variety of human shapes, poses, actions and virtual backgrounds to support deep learning systems.

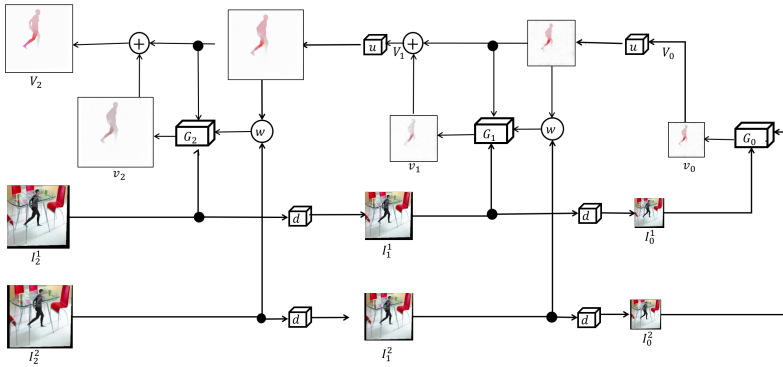


Figure 3: A Spatial Pyramid Network [55] for Optical Flow. At each pyramid level, network G_k predicts flow residuals v_k that get added up to produce full flow V_2 . w is warping operator. u, d are learned convolutional layers that upsample flows and downsample images. The figure shows 3 pyramid levels for simplicity. Our implementation uses 4 levels.

4 Learning

Our neural network derives from SPyNet [55], which employs different convnets at different levels of an image pyramid. In SPyNet, these convnets are trained independently and sequentially to estimate optical flow. In contrast, we introduce learnable convolutional layers u, d , for upsampling and downsampling between pyramid levels. In SPyNet, these are fixed bilinear operators. A differential warping operator, w [24] facilitates our model to be fully differentiable and end-to-end trainable. Thus, we perform joint training of all the convnets at different pyramid levels rather than training them sequentially like SPyNet. We briefly describe the spatial pyramid structure and introduce our network and learning process below.

Our architecture consists of 4 pyramid levels. For simplicity, we show 3 of the 4 pyramid levels in Figure 3. Each level works on a particular resolution of the image. The top level works on the full resolution and the images are downsampled as we move to the bottom of the pyramid. Each level learns a convolutional layer d , to perform downsampling of images. Similarly, a convolution layer u , is learned for upsampling optical flow. At each level, we also learn a convnet G_k to predict optical flow residuals v_k at that level. These flow residuals get added at each level to produce the full flow, V_K at the finest level of the pyramid.

Each convnet G_k takes a pair of images as inputs along with flow V_{k-1} obtained by upsampling the output of the previous level. The second frame is however warped using V_{k-1} and the triplet $\{I_k^1, w(I_k^2, V_{k-1}), V_{k-1}\}$ is fed as input to the convnet G_k . The structure of the convnets G_k is same as SPyNet. Each of the convnets G_k is a five layer network containing $\{32, 64, 32, 16, 2\}$ feature maps with 7×7 kernels. At each level, the downsampling layers d learn 3×3 convolutional kernels with 6 feature maps to operate on 6 channels of image pairs. Similarly, the upsampling layers u learn 4×4 convolutional kernels with 2 feature maps to operate on 2-channel flows. We use w to refer to a bilinear warping operator which is non-learnable. The general structure of spatial pyramids can be seen in [55]. We import the weights of our convnets G_i from the first four convnets $\{G_0, G_1, G_2, G_3\}$ of SPyNet pre-trained on the Flying Chairs dataset [9]. We use a differentiable warping operator, w [24]. We now construct our fully differentiable spatial pyramid architecture and train it end-to-end minimizing an End Point Error (EPE).

Hyperparameters. We use Adam [24] to optimize our loss at a constant learning rate of 10^{-6} , $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We use a batch size of 8 and run 4000 iterations per epoch. We train our model for 100 epochs on the Human Flow dataset. We use the Torch³ framework for our implementation and use four Nvidia K80 GPUs to train in parallel. It takes 1 day for our model to train.

Data Augmentations. We also augment our data by applying several transformations and adding noise. Although our dataset is quite large, augmentation improves the quality of results on real scenes. In particular, we apply scaling in the range of $[0.3, 3]$, and rotations in $[-17^\circ, 17^\circ]$. We randomly crop the images and flows to 256×256 at the finest level. We sample uniform Gaussian noise $\mathcal{N}(0, 1)$ and add it to the images with a weight factor of 1:10. We also apply color jitter with additive brightness, contrast, and saturation changes. The dataset is normalized to have zero mean and unit standard deviation using [24].

5 Experiments

We compare the average End Point Errors (EPEs) of competing methods in Table 1(b) along with the time for evaluation. Human motion is complex and general optical flow methods fail to capture it. Our trained network outperforms previous methods, and SPyNet [35] in particular, in terms of average EPE on the Human Flow Dataset. This indicates that other optical flow networks can also improve their performance on human motion using our dataset. We show visual comparisons in Figure 4.

Optical flow methods that employ learning using large datasets such as FlowNetS [9] show poor generalization on our dataset. Since the results of FlowNetS are very close to the zero flow baseline (Table 1), we cross-verify by evaluating FlowNetS on a mixture of Flying Chairs [9] and Human Flow dataset and observe that the flow outputs on human flow dataset is quite random (see Figure 4). The main reason is that our dataset contains a significant amount of small motions and it is known that FlowNetS does not perform very well on small motions. SPyNet [35] however performs quite well and is able to generalize to body motions. The results however look noisy in many cases.

Our dataset employs a layered structure where a human is placed against a background. As such layered methods like PCA-layers [42] perform very well on a few images (row 4 in Figure 4) where they are able to segment a person from the background. However, in most cases, they do not obtain good segmentation into layers.

Previous state-of-the-art methods like LDOF [4], EpicFlow [36] and FlowFields [10] perform much better than others. They get a good overall shape, and smooth backgrounds. However, their estimation is quite blurred. They tend to miss the sharp edges that are typical of human hands and legs. They are also significantly slower.

In contrast, our network outperforms the state of the art optical flow methods by 30% on the Human Flow dataset. Our qualitative results show that our method can capture sharp details like hands and legs of the person. Since, our test data is comparatively large, we evaluate only the fastest optical methods on our dataset.

Real Scenes. We show a visual comparison of results on real-world scenes of people in motion. We collect these scenes by cropping people from real world videos as shown in Figure 5(a). We use DPM [44] for detecting people and compute bounding box regions in two frames using the ground truth of the MOT16 dataset [32].

³<http://torch.ch>

Dataset	# Train	# Test	Resolution			
	Frames	Frames		Method	AEPE	Time(s)
MPI Sintel[5]	1,064	564	1024 × 436	Zero	0.6611	-
KITTI 2012[12]	194	195	1226 × 370	FlowNet[9]	0.5846	0.080
KITTI 2015[6]	200	200	1242 × 375	PCA Flow[12]	0.3652	10.357
Virtual Kitt[15]	21,260	—	1242 × 375	SPyNet[15]	0.2066	0.038
Flying Chairs[9]	22,232	640	512 × 384	Epic Flow[16]	0.1940	1.863
Flying Things[13]	21,818	4,248	960 × 540	LDOF[10]	0.1881	8.620
Monkaa[14]	8,591	—	960 × 540	Flow Fields[11]	0.1709	4.204
Driving[13]	4,392	—	960 × 540	HumanFlow	0.1164	0.031
Human Flow	135,153	10,867	256 × 256		(b)	

(a)

Table 1: (a) Comparison of Human Flow dataset with previous optical flow datasets. (b) EPE comparisons and evaluation times of different optical flow methods on the Human Flow dataset. Zero refers to the EPE when zero flow is always used for evaluation.

We visually compare our results with other popular optical flow methods on real world scenes in Figure 5(b). The performance of PCA-Layers [12] is highly dependent on its ability to segment. Hence, we see only a few cases where it looks visually correct. SPyNet [15] gets the overall shape but the results look noisy in certain image parts. While LDOF [10], EpicFlow [16] and FlowFields [11] generally perform well, they often find it difficult to resolve the legs, hands and head of the person. The results from HumanFlow look appealing especially while resolving the overall human shape, and various parts like legs, hands and the human head. HumanFlow also performs very well under occlusions when the full body is not visible.

Timing Evaluation. Although some of the methods do quite well on our benchmark, they tend to be slow due to their complex nature. As such, they are not likely to be used in real time or embedded applications. We show timing comparisons on a pair of frames in Table 1. We show that learning with human flow data can make our model a lot simpler while simultaneously making it fast and accurate for capturing human motion. Our model takes 31 ms for inference on NVIDIA TitanX. As such it can run in real time at 32 fps.

Network Size. Employing spatial pyramid structures [15] to neural network architectures leads to significant reduction of size. As such our network is quite small and can be stored in 7.8 MB of memory. Our network has a total of 4.2 million learnable parameters making it fast and easy to train.

6 Conclusion and Future Work

In summary, we created an extensive dataset containing images of realistic human shapes in motion together with groundtruth optical flow. The realism and extent of this dataset, together with an end-to-end trained system, allows our new HumanFlow method to outperform previous state-of-the-art optical flow methods in our new human-specific dataset. Furthermore, we show that our method generalizes to human motion in real world scenes for optical flow computation. Our method is compact and runs in real time making it highly suitable for phones and embedded devices.

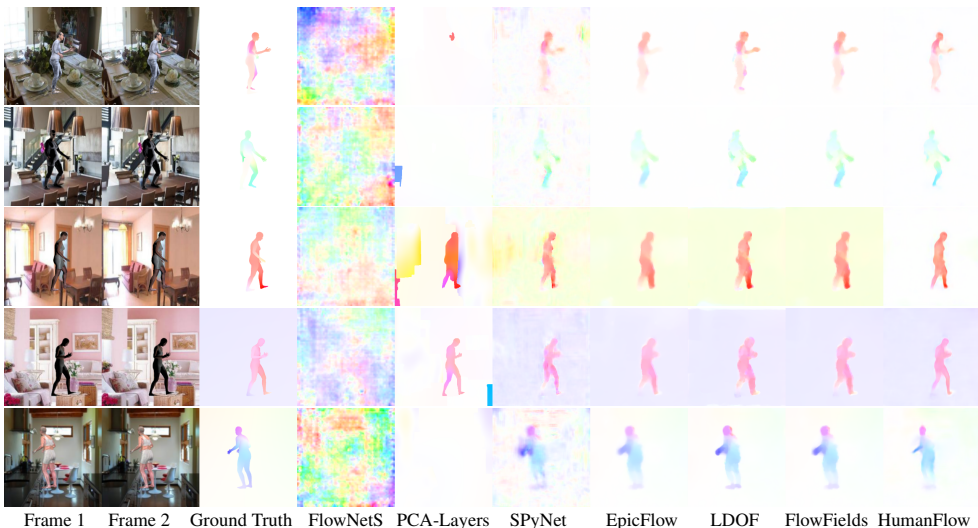


Figure 4: Visual comparison of optical flow estimates using different methods on the Human Flow test set. From left to right, we show Frame 1, Frame 2, Ground Truth flow, results on FlowNetS [9], PCA-Layers [22], SPyNet [55], EpicFlow [56], LDOF [4], FlowFields [10] and HumanFlow (ours).

In future work, we plan to model more subtle human motions such as those of faces and hands. We also plan to generate training sequences containing multiple, interacting people, and more complex 3D scene motions. Additionally, we plan to add 3D clothing and accessories. The dataset and our focus on human flow opens up a number of research directions in human motion understanding and optical flow computation. We would like to extend our dataset by modeling more diverse clothing and outdoor scenarios. A direction of potentially high impact for this work is to integrate it in end-to-end systems for action recognition, which typically take pre-computed optical flow as input. The real-time nature of the method could support motion-based interfaces, potentially even on devices like cell phones with limited computing power. The data, model, and training code is available, enabling researchers to apply this to problems involving human motion.



Figure 5: (a) We use a person detector to crop out people from real-world scenes (left) and use our model to compute optical flow on the cropped section (right). (b) Visual comparison on real scenes. From left to right, we show Frame 1, Frame 2, results on PCA-Layers [22], and SPyNet [55], EpicFlow [56], LDOF [4], FlowFields [10] and HumanFlow (ours). Our method is good at picking up limb and head motion.

Acknowledgements

We thank Siyu Tang for compiling the person detections and Yiyi Liao for helping us with optical flow evaluation. We thank Cristian Sminchisescu for the Human3.6M mocap marker data.

References

- [1] Christian Bailer, Bertram Taetz, and Didier Stricker. Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4015–4023, 2015.
- [2] Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, 2011.
- [3] M. J. Black, Y. Yacoob, A. D. Jepson, and D. J. Fleet. Learning parameterized models of image motion. In *IEEE Conf. on Computer Vision and Pattern Recognition, CVPR-97*, pages 561–567, Puerto Rico, June 1997.
- [4] Thomas Brox, Christoph Bregler, and Jitendra Malik. Large displacement optical flow. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 41–48. IEEE, 2009.
- [5] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In A. Fitzgibbon et al. (Eds.), editor, *European Conf. on Computer Vision (ECCV)*, Part IV, LNCS 7577, pages 611–625. Springer-Verlag, October 2012.
- [6] James Charles, Tomas Pfister, Derek R. Magee, David C. Hogg, and Andrew Zisserman. Personalizing human video pose estimation. In *CVPR*, pages 3063–3072. IEEE Computer Society, 2016. ISBN 978-1-4673-8851-1. URL <http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7776647>.
- [7] Sun D., S Roth, JP Lewis, and MJ Black. Learning optical flow. In *ECCV*, pages 83–97, 2008.
- [8] J. W. Davis. Hierarchical motion history images for recognizing human motion. In *Detection and Recognition of Events in Video*, pages 39–46, 2001. URL <http://dx.doi.org/10.1109/EVENT.2001.938864>.
- [9] Alexey Dosovitskiy, Philipp Fischery, Eddy Ilg, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, Thomas Brox, et al. Flownet: Learning optical flow with convolutional networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2758–2766. IEEE, 2015.
- [10] R. Fablet and M. J. Black. Automatic detection and tracking of human motion with a view-based representation. In *European Conf. on Computer Vision, ECCV 2002*, volume 1 of LNCS 2353, pages 476–491. Springer-Verlag, 2002.

- [11] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, pages 1933–1941. IEEE Computer Society, 2016. ISBN 978-1-4673-8851-1. URL <http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7776647>.
- [12] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 32(9):1627–1645, 2010.
- [13] K. Fragkiadaki, H. Hu, and J. Shi. Pose from flow and flow from pose. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2059–2066, June 2013. doi: 10.1109/CVPR.2013.268.
- [14] William T. Freeman, Egon C. Pasztor, and Owen T. Carmichael. Learning low-level vision. *International Journal of Computer Vision*, 40(1):25–47, 2000. ISSN 1573-1405. doi: 10.1023/A:1026501619075. URL <http://dx.doi.org/10.1023/A:1026501619075>.
- [15] A Gaidon, Q Wang, Y Cabon, and E Vig. Virtual worlds as proxy for multi-object tracking analysis. In *CVPR*, 2016.
- [16] Adrien Gaidon, Zaid Harchaoui, and Cordelia Schmid. Activity representation with motion hierarchies. *International Journal of Computer Vision*, 107(3):219–238, 2014. ISSN 1573-1405. doi: 10.1007/s11263-013-0677-1.
- [17] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [18] Donald Geman and Stuart Geman. Opinion: Science in the age of selfies. *Proceedings of the National Academy of Sciences*, 113(34):9384–9387, 2016.
- [19] Robin Green. Spherical Harmonic Lighting: The Gritty Details. *Archives of the Game Developers Conference*, March 2003. URL <http://www.research.scea.com/gdc2003/spherical-harmonic-lighting.pdf>.
- [20] Fatma Güney and Andreas Geiger. Deep discrete flow. In *Asian Conference on Computer Vision (ACCV)*, 2016.
- [21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.
- [22] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. FlowNet 2.0: Evolution of optical flow estimation with deep networks. *arXiv preprint arXiv:1612.01925*, 2016.
- [23] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, jul 2014.
- [24] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015.

- [25] Hueihan Jhuang, Juergen Gall, Silvia Zuffi, Cordelia Schmid, and Michael J. Black. Towards understanding action recognition. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3192–3199, Sydney, Australia, December 2013. IEEE.
- [26] Gunnar Johansson. Visual perception of biological motion and a model for its analysis. *Perception & Psychophysics*, 14(2):201–211, 1973. ISSN 1532-5962. doi: 10.3758/BF03212378.
- [27] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [28] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2556–2563. IEEE, 2011.
- [29] Matthew Loper, Naureen Mahmood, and Michael J Black. Mosh: Motion and shape capture from sparse markers. *ACM Transactions on Graphics (TOG)*, 33(6):220, 2014.
- [30] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, October 2015.
- [31] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3061–3070, 2015.
- [32] Anton Milan, Laura Leal-Taixé, Ian D. Reid, Stefan Roth, and Konrad Schindler. MOT16: A benchmark for multi-object tracking. *arXiv:1603.00831*, 2016.
- [33] N.Mayer, E.Ilg, P.Häusser, P.Fischer, D.Cremers, A.Dosovitskiy, and T.Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. URL <http://lmb.informatik.uni-freiburg.de/Publications/2016/MIFDB16>. arXiv:1512.02134.
- [34] Tomas Pfister, James Charles, and Andrew Zisserman. Flowing convnets for human pose estimation in videos. In *ICCV*, pages 1913–1921. IEEE Computer Society, 2015. ISBN 978-1-4673-8391-2. URL <http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7407725>.
- [35] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. *arXiv preprint arXiv:1611.00850*, 2016.
- [36] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. EpicFlow: Edge-Preserving Interpolation of Correspondences for Optical Flow. In *Computer Vision and Pattern Recognition*, 2015.
- [37] Kathleen M Robinette, Sherri Blackwell, Hein Daanen, Mark Boehmer, and Scott Fleming. Civilian american and european surface anthropometry resource (caesar), final report. volume 1. summary. Technical report, DTIC Document, 2002.

- [38] Javier Romero, Matthew Loper, and Michael J. Black. FlowCap: 2D human pose from optical flow. In *Pattern Recognition, Proc. 37th German Conference on Pattern Recognition (GCPR)*, volume LNCS 9358, pages 412–423. Springer, 2015.
- [39] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.
- [40] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Deep End2End Voxel2Voxel prediction. In *The 3rd Workshop on Deep Learning in Computer Vision*, 2016.
- [41] Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *CVPR*.
- [42] Jonas Wulff and Michael J Black. Efficient sparse-to-dense optical flow estimation using a learned basis and layers. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 120–130. IEEE, 2015.
- [43] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv:1506.03365*, 2015.
- [44] Silvia Zuffi, Javier Romero, Cordelia Schmid, and Michael J Black. Estimating human pose with flowing puppets. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3312–3319, 2013.