FlowCap: 2D Human Pose from Optical Flow

Javier Romero, Matthew Loper, Michael J. Black

Max Planck Institute for Intelligent Systems, Tübingen, Germany

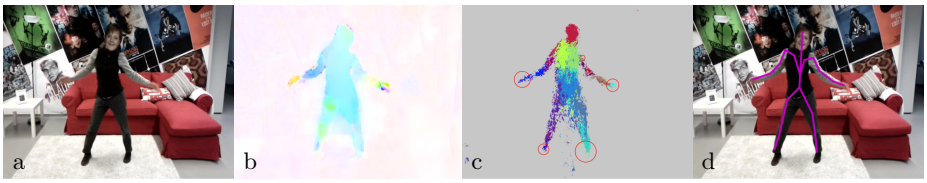

Fig. 1: **FlowCap overview. a.** Example frame from a video sequence shot with a phone camera. **b.** Optical flow computed with GPU flow [1]. **c.** Per-pixel part assignments based on flow with overlaid uncertainty ellipses (red). **d.** Predicted 2D part centroids connected in a tree.

Abstract. We estimate 2D human pose from video using *only optical flow*. The key insight is that dense optical flow can provide information about 2D body pose. Like range data, flow is largely invariant to appearance but unlike depth it can be directly computed from monocular video. We demonstrate that body parts can be detected from dense flow using the same random forest approach used by the Microsoft Kinect. Unlike range data, however, when people stop moving, there is no optical flow and they effectively disappear. To address this, our *FlowCap* method uses a Kalman filter to propagate body part positions and velocities over time and a regression method to predict 2D body pose from part centers. No range sensor is required and FlowCap estimates 2D human pose from monocular video sources containing human motion. Such sources include hand-held phone cameras and archival television video. We demonstrate 2D body pose estimation in a range of scenarios and show that the method works with real-time optical flow. The results suggest that optical flow shares invariances with range data that, when complemented with tracking, make it valuable for pose estimation.

1 Introduction

Human pose estimation from monocular video has been extensively studied but currently there are no widely available, general, efficient, and reliable solutions. The problem is challenging due to the dimensionality of articulated human pose, the complexity of human motion, and the variability of human appearance in images due to clothing, lighting, camera view, and self occlusion. There has been extensive work on 2D human pose estimation using part-based models [8, 11,

12, 19, 27, 29], but existing solutions are still brittle. Systems like the Microsoft Kinect [21] address the above issues by using a specialized depth sensor that simplifies the problem by exploiting additional information. Depth data enables direct estimation of 3D pose while providing invariance to appearance.

What is missing is a robust solution like Kinect for the general 2D human pose estimation problem from video; that is, one that applies to archival video sources and can be used with devices such as cell phones and laptops that are currently equipped only with a monocular video camera. We propose optical flow as a key ingredient for such a solution, and demonstrate its potential with a system called *FlowCap* that estimates 2D pose using only optical flow.

Our method is made possible by the following observation: *Optical flow contains much of the same information as range data.* An optical flow field is much like a depth map in that the effects of appearance are essentially removed (see **Supp. Mat.**). Flow captures information about the overall shape and pose of the body and the boundary between the body and the background (Fig. 1b). Moreover, flow has an advantage beyond range data: 2D flow also captures the *motion* of body parts and we use this to good effect.

The first component of our approach follows that of Shotton et al. [21] except we replace range data with optical flow. We train a regression forest using flow and body part segmentations of realistic synthetic bodies in motion. As in [21] we predict per-pixel body part assignments and identify the part centroids (Fig. 1c).

Optical flow has one key disadvantage relative to range data: When a person is stationary, flow does not tell us where they are. It does however tell us something important – that the person is not moving. To take advantage of this, the second component of our method adds a temporal prediction process on top of the body part detections. We use a Kalman filter to estimate the locations and velocities of all body parts in 2D. By estimating velocities, we are able to incorporate information from the optical flow into the Kalman observation model. This improves part estimation when the person is moving as well as when they are still. When a person stops moving, the flow is near zero and the Kalman filter predicts the body is not moving, resulting in a stable pose estimate.

Using the HumanEva benchmark [23] we compare FlowCap with a state-of the-art single-frame method [27] and find that, when people are moving, FlowCap is more stable. We demonstrate that the accuracy of real time optical flow estimation (GPU4Vision [26]) is sufficient for our task. We also test FlowCap on video sequences captured outdoors, with a moving hand-held cell-phone camera, and with archival video from television.

We do not propose FlowCap as a complete, stand-alone, system. Our approach, using only flow, cannot compete with Kinect's use of range data for accuracy or for 3D estimation. Rather our goal is to show that optical flow has a role in human pose estimation and that it shares properties with depth data. Clearly a full solution will include color data but here we demonstrate how far one can get with flow alone. To facilitate further work, we will make our training set of flow data available for research purposes[1].

[1] http://ps.is.tuebingen.mpg.de/project/FlowCap

2 Prior Work

There is a huge literature on pose estimation in static images, video sequences, and using depth information from many sources. We focus on 2D human pose, which is widely studied and useful for applications such as person detection, human tracking, activity analysis, video indexing, and gesture recognition. Here we focus on the two areas most closely related to our method: Microsoft's Kinect and articulated pose estimation from optical flow.

Kinect: Kinect performs human motion capture from an inexpensive device in a person's home with sufficient accuracy for entertainment purposes. While popular, range sensing devices like Kinect are still not widely deployed when compared with traditional video cameras. Since the Kinect works only on range data it cannot be used for human pose estimation with archival data from television and films. Additionally, the Kinect's IR illumination can be swamped by natural light, rendering it useless outside.

One key to the success of Kinect is the use of regression forests [21]. Unfortunately, it is not feasible to apply this method directly to regular video images due to the huge variability in human appearance. Range data is important for the success of Kinect for two reasons. First it provides direct observations of scene depth, removing the ambiguities inherent in the 2D projection of people onto the image plane of a monocular camera. Second, and just as important, is that the range data simplifies the signal processing problem by removing the irrelevant effects of appearance and lighting while maintaining the important information about body structure. Our observation is that optical flow provides similar benefits, in particular with respect to this second point.

The first step of our method uses the regression forest of [21] but replaces depth training data with optical flow. After this we deviate from [21] because, unlike range, when the person stops moving the flow is zero. Consequently to know where the person is, our method requires a temporal model to integrate information; [21] does not use a temporal model but rather, finds the person again in every frame.

Pose from flow: There are many 2D and 3D model-based methods for estimating human pose from video that exploit optical flow (e.g. [6, 17, 22, 24]). These methods relate the 2D image motion to the parameters of an articulated figure. Motion History Images [5] have also been used for pose classification.

Fablet and Black [10] use a synthetic character and motion capture data to generate training flow fields from different views. They use PCA to construct a low-dimensional representation of the flow and represent simple activities as trajectories in that low-dimensional space. They use a multi-view representation to cope with changing 3D viewpoint but do not estimate articulated pose.

Efros et al. [7] use optical flow patterns to estimate pose. They focus on low resolution people in video, which makes the flow information limited and noisy. Consequently they treat it as a spatio-temporal pattern, which becomes a motion descriptor, used to query a database for the nearest neighbor with a similar pattern and known 2D and 3D pose. They require similar sequences of full body poses in the database.

Bissacco et al. [3] train a boosted regression method to recognize pose from image and motion features. They do not use optical flow directly, but rather work on image differences. Schwarz et al. [20] use flow between time of flight range images to help differentiate body parts that occlude each other but do not estimate body pose from flow.

Recently, several methods augment traditional 2D pose estimation with optical flow information. In [13] they use flow to help segment body parts while jointly reasoning about pose, segmentation, and motion. In [29] they use flow to propagate putative 2D body models to neighboring frames. This enables an image likelihood function that incorporates information from multiple frames. In [16] the authors train a deep convolutional neural network (CNN) to use images and flow to estimate upper body pose. These approaches rely primarily on non-flow image cues, with flow as an extra cue. Here we explore the question of how far we can go with flow alone.

3 Data

Like [21] we generate training data using a realistic 3D human body model. However, generating a good flow training set, differs from their approach. First, the same body pose at time t can move to many different poses at $t+1$ resulting in different flow fields. Consequently, the training data must cover a range of both poses and changes in pose. Second, camera motions change the observed flow. While we robustly estimate and remove camera motion we assume there will be some residual camera motion and consequently build this into our training set to improve robustness. Third, optical flow computed on real images is not perfect and can be affected by lighting, shadows, and image texture (or lack thereof); we need to realistically model this noise. To do so, we synthesize pairs of frames with varied foreground and background texture, and various types of noise, and then run a flow algorithm to compute the training flow. The training dataset contains realistic human bodies in varying home environments performing a variety of movements. Example training data is shown in Fig. 2a.

Body shape variation. We use a 3D body model [15] that allows us to generate 3D human bodies with realistic shapes in arbitrary poses. We use separate body shape models for men and women and generate a wide variety of body shapes. The model represents people in tight clothing, but future work could add synthetic clothing and hair.

As in [21], the body model is segmented into parts, which are color coded for visualization (Fig. 2a bottom). The training data includes the 2D projection of these part segments and the 2D centroids of each part. Note that we use 19 parts, fewer and larger than in [21]; these provide more reliable part detection.

Body pose variation. To capture a wide range of human poses and motions we generate training pairs of poses representing plausible human movements between two frames. We do this in two ways. For experiments with the HumanEva dataset, we take the motion capture data from the training set and animate bodies using these motions. While appropriate for the HumanEva evaluation, the

set of motions is somewhat limited. Consequently for our other experiments, we create a *generic* motion dataset. We create a distribution of natural poses from a dataset of 3D registrations like [4]. Then we sample pairs of poses and generate paths between them in pose space. Finally, we sample points along these paths, biased towards one of the originals, to define the pose change between frames.

Appearance variation. The performance of optical flow methods is affected by image texture and contrast. For example, when the background is homogeneous the estimated optical flow field may be overly smooth, blurring the foreground motion with the background motion; this can be clearly seen in Fig. 2a. We posit that these effects should be present in our dataset to be able to successfully estimate human pose from real flow.

We created high resolution texture maps from 3D scans of over 30 subjects. For each body shape, we randomly select a texture map and render the body in a basic 3D environment with a wall, floor, some simple objects, and some independently moving objects to simulate clutter and background motion. While not photo-realistic, the scenes have relatively realistic lighting, blur, and noise.

Flow computation. Flow algorithms make different trade-offs between accuracy and speed. To evaluate whether the real-time estimation of 2D body pose is feasible, we compare two methods using [1]: one non-real-time (3 seconds/frame) and the other real-time but noisier. For the former we use the Huber-L1 method from [26]. For the latter we use FAST_HL1 in [1].

Scale variation. As is common in the 2D human pose literature, we train two separate models at different scales (Fig. 2a left and right). The appropriate model is manually picked depending on the test sequence. The first captures upper body movement common in archival video like TV sitcoms. The second captures the full body and is aimed at game applications like in [21]. Within each category we generate training samples with a range of scales to provide some scale invariance. This scale invariance is demonstrated in our experiments with HumanEva, in which the size of the person varies substantially.

Training data summary. The HumanEva training set is composed of approximate $7,000$ training examples of the full body. We generate two generic datasets: The upper body dataset is composed of approximately $7,000$ training examples, while the full body dataset has approximately $14,000$.

4 Method

The goal is to sequentially estimate the 2D pose of a human body from a series of images. As in [21], we consider two subproblems: A classification problem of assigning a body part identifier to each pixel, and a regression problem of inferring the position of the body joints. We add an additional tracking component that is essential when using flow.

Problem definition: Our input consists of a sequence of $k+1$ images, Y_i, of dimensions $m \times n$. For each image Y_i, we estimate the optical flow field, V_i, between Y_i and Y_{i+1} as described in Sec. 3. To reduce the effect of camera motion we also robustly estimate a dominant global homography for the image

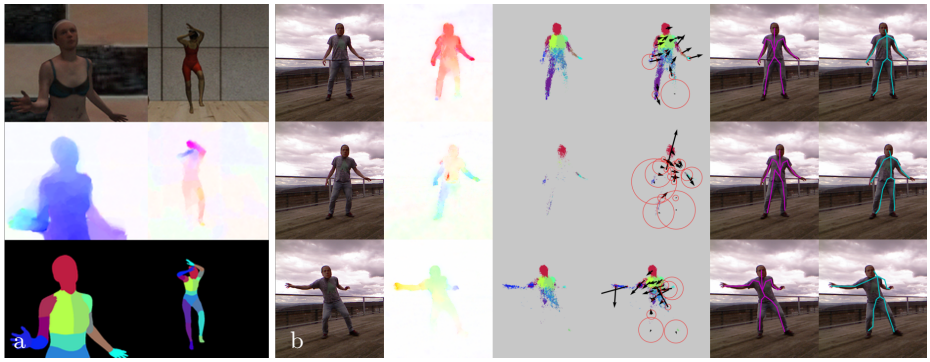

Fig. 2: **a) Training data.** Top row: example synthetic frames from pairs of training frames. Middle: Estimated optical flow for each frame. Bottom: Ground truth body part segmentations. **b) Visual summary of the method.** Left to right: image capture with Kinect RGB camera, optical flow (color coded as in [2]), per pixel part labels, part centers with uncertainty (red circles) and motion vectors (10x actual magnitude), estimated kinematic structure of the part centers, predicted Kinect kinematic structure using linear regression.

pair using RANSAC. Let the flow field at every pixel, given by the homography, be H_i. Then we define the residual flow field to be $\hat{V}_i = V_i - H_i$.

For every residual flow field, $\hat{V}_i$, our goal is to estimate the 2D locations of j joints, X_i, of size $j \times 2$; like [21], we use body part assignments to p parts as an intermediary between observables and joint locations. This is achieved in three steps. First, we estimate per-pixel body part assignments with a matrix, P_i, of size $m \times n \times (p+1)$; labels correspond to either one of p body parts or the background. A label matrix, L_i, of size $m \times n$ is simply computed as $L_i(\mathbf{x}) = \arg\max_l P_i(\mathbf{x}, l)$, where $\mathbf{x} = (x, y)$ is an image pixel location. Second, we compute a matrix, M_i, of size $p \times 2$ containing the 2D centroids of the body parts in the image. Finally, the matrix, X_i, of 2D joint locations is predicted from M_i using linear regression.

Flow difference features: Following [21], each pixel is described by a t dimensional feature vector $F_i(\mathbf{x})$. Here we take $F_i(\mathbf{x})$ to include the flow magnitude $\|\hat{V}_i(\mathbf{x})\|$ at the pixel and a set of $t-1$ flow differences, $\|\hat{V}_i(\mathbf{x}) - \hat{V}_i(x+\delta_x, y+\delta_y)\|$, computed with random surrounding pixels. The maximum displacements, δ_x, δ_y are set to 160 pixels for the full body training set and 400 pixels for the upper body set. A full body typically occupies around 100×300 pixels. Inspired by [21], we chose $t = 200$ and draw the samples δ_x, δ_y from a Gaussian distribution.

Body part classification: FlowCap classifies each feature vector, $F_i(\mathbf{x})$, at each pixel into one of $p+1$ classes representing the p body parts and the background. Randomized decision forests (implementation from [18]) are used to classify flow difference features. For each training image, we randomly sample 2000 pixel locations uniformly per part and use the associated feature vectors to train the classifier. Six trees are trained with maximum depth so that leaves

contain a minimum of four samples. Given a flow field as input, the output of the decision forest is a matrix P_i from which we compute the label matrix L_i.

In the absence of motion, the classification, L_i, is ambiguous (row 2 in Fig. 2b). A static pixel surrounded by static pixels is classified as background. However, the lack of motion is a strong, complementary, feature that we can exploit in a tracking scheme. In this way, optical flow is used in two ways: first, as a static, appearance-invariant, feature for per-frame pose estimation, and second, as an observation of the pixel velocities for effective part tracking.

Part centroid tracking: The per-pixel part classifications are now used to track the part positions. For simplicity, we track a single hypothesis $\hat{M}_i(l)$ of the centroid of each part l. Considering multiple modes is promising and left for future work. While the most straightforward estimation of the 2D centroids would be a weighted average according to probabilities P_i, we seek a more robust estimation based on the following approximation of the mode

$$\hat{M}_i(l) = \sum_{\mathbf{x}} P_i(\mathbf{x}, l)^\alpha \mathbf{x} / \sum_{\mathbf{x}} P_i(\mathbf{x}, l)^\alpha \tag{1}$$

where $\alpha = 6$ in our experiments. Alternatively, this could be done by retraining the regression tree leaves to infer pixel offsets to the joint centroids, $\hat{M}_i - \mathbf{x}$ [21].

The modes can be very inaccurate in the absence of movement. To address this we perform temporal tracking of the centroids (independently per part) using a linear Kalman filter [25]. The state of the filter contains the estimation of the position and velocity of each part centroid, $M_i(l), M'_i(l)$. The measurements are the centroid estimates, $\hat{M}_i(l)$, and the velocities, $\hat{M}'_i(l)$, which we compute from the optical flow in a region around the current estimate. Since we are directly observing estimations of our state, the observation model is the identity. The states are initialized with their corresponding measurement $M_0(l) = \hat{M}_0(l)$, $M'_0(l) = \hat{M}'_0(l)$. The state-transition model assumes constant velocity:

$$M_i(l) = M_{i-1}(l) + M'_{i-1}(l) \tag{2}$$

$$M'_i(l) = M'_{i-1}(l). \tag{3}$$

The definition of the process and measurement noise is not so straight-forward. All noise models are considered uncorrelated. We empirically set the transition noise standard deviations to values between 2 and 20 pixels depending on the body part. The velocity component of the measurement noise, related to the flow accuracy, is empirically set to standard deviations of 5 pixels. The position component of the measurement noise Q_i^M depends on the accuracy of the decision forest estimation, through its estimations of the per-part probability matrices P_i

$$Q_i^M(l) = k_i^2 / \Big(\sum_{\mathbf{x}} P_i(\mathbf{x}, l) \Big)^2 \tag{4}$$

where k_i is a part-dependent constant, with empirical values between 40 and 100 pixels, reflecting the accuracy differences of the random forest across body parts.

Predicting joints: Tracking results in estimations of the body part centroids, M_i; Fig. 2b, fifth column, shows estimated part centers connected by purple lines. For many applications, however, we want the locations of the joints in an articulated model. One could directly learn these using the regression forest but it is more straightforward to estimate part centers and then estimate joint locations from these.

The relation between part centroids and joint locations is learned from the training dataset described in Sec. 3. Joint positions are predicted linearly from centroids, both represented in 2D. On HumanEva training data, we regress from detected part centroids to the ground truth 2D marker locations with an L1 loss. For the other experiments we use the generic training data and train the regression function from the ground truth part centroids to the ground truth model joints using elastic net [28]. Figure 2b, sixth column, shows the kinematic tree corresponding to predicted joints in turquoise.

5 Experiments

We summarize the experiments here; see supplemental video for more.

1. HumanEva. We compare FlowCap's performance on monocular 2D human pose estimation with [27]. This single-frame method estimates human pose based on the image gradients. In contrast, FlowCap completely disregards the visual appearance of a single frame, exploiting solely optical flow. The comparison is performed on the validation set of HumanEva I [23], which contains sequences of multiple subjects performing a variety of actions. We evaluate the methods on video from the single color camera, C1, for sequences containing movement for every body part, namely "Walking" and "Jog". The motions involve significant changes in scale and a full 360 degree change in orientation of the body.

Figure 3a shows 2D marker error, and confirms that FlowCap outperforms [27] on this subset of HumanEva I. The method of [27] has large errors in some frames due to misdetections on the background or large errors of the arm joints. This is reflected in larger standard deviations. While not a comprehensive comparison, this suggests flow can be a useful cue for 2D human pose.

2. Outdoors. While Kinect works well indoors, we captured a game-like sequence outside using the Kinect camera (Fig. 2). The natural lighting causes Kinect pose estimation to fail on almost every frame of the sequence. In contrast, FlowCap recovers qualitatively good 2D pose.

3. Cellphone camera. A truly portable system for human pose estimation would open up many applications. Figure 4a shows FlowCap run on video from a hand-held Samsung Nexus S mobile phone. Despite the camera-motion removal step (Sec. 4), residual background flow is observable in the sequence. Nonetheless, the estimated 2D poses are qualitatively good. This is a proof of concept since our software is not designed to run on a phone and all processing is done off-line.

4. Television. We do not claim to have a complete solution for human pose estimation from archival data but Fig. 4b shows a few results on the TV series "Buffy the Vampire Slayer" and "Friends." Results on videos with mostly-frontal

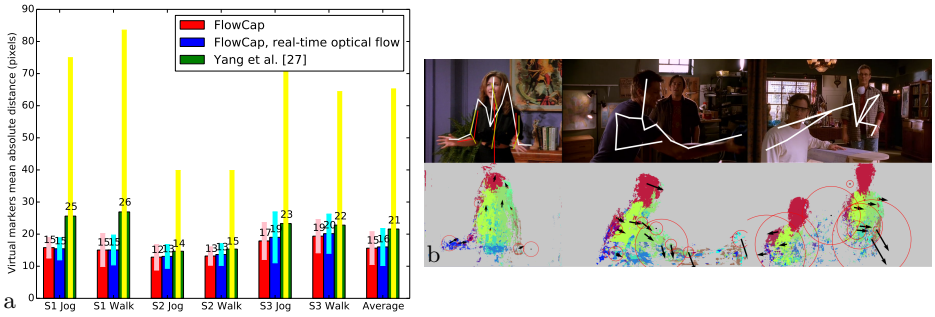

Fig. 3: **a) Ground truth evaluation.** Average (and std) absolute marker distance (as in [23]) for Walking and Jog validation sequences in HumanEva for FlowCap, FlowCap with real-time flow and [27]. **b) Failure cases.** Lack of representative data (e.g. long hair), back person's view, and multiple people.

views of a single moving person are promising. Here we envision FlowCap as part of a more complex system using multiple cues or as an initialization to a part-based model like [27].

Running time. Here we have shown a proof of concept system. Each component of FlowCap is either real-time now or could be realistically made real time (flow estimation, part prediction, Kalman filtering, and pose estimation). The optical flow method of [26] used in most of the experiments has a running time of 3 seconds in a Nvidia Quadro K4000. We also experimented with a fast version of the flow code that runs at about 30ms/frame. Despite lower quality flow, the results in Fig. 3a show that FlowCap performance degrades very little when using the real-time optical flow. Flow feature extraction and the Random Forest method are slow; currently taking on the order of ten seconds per frame in VGA images. However, these can run in super-real-time [21]. The running times of our Kalman Filter and of the regression to joint space are negligible.

Failure cases and future work. Although we have shown that our system works well in a number of situations, there is still room for improvement. Fig. 3b shows that our system would benefit from improving the realism of training data, better disambiguation between front and back poses or tracking multiple subjects. An obvious drawback of using only flow is that our system only tracks body parts that have moved in the past; this could be solved by using image-based initialization. Other future directions include a multi-camera version [9], model-based tracking, dealing with background motions and using multi-frame optical flow features. More sophisticated flow algorithms could also be evaluated.

6 Conclusion

We have demonstrated how optical flow alone can provide information for 2D human pose estimation. Like range data, it can factor out variations in image appearance and additionally gives information about the motion of body parts. We have also demonstrated how flow can be used to detect and track human pose

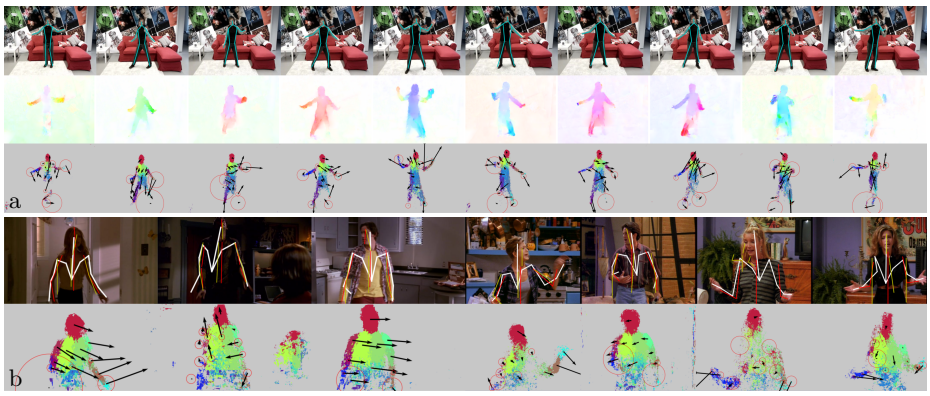

Fig. 4: **a) Smartphone FlowCap.** Here the video is captured using a hand-held phone camera. This results in overall flow due to rotation and translation of the camera. Despite this, part estimates remain good and pose is well estimated. **b) Archival video.** Results on archival data from series Friends and Buffy. Ground truth shown in red, [27] in yellow, FlowCap part centers in white.

in monocular videos such as television shows. This demonstrates a simple proof of the concept that flow offers something like the appearance invariance of depth while being available from ordinary video. The application of the techniques from [21] to monocular flow fields is non-obvious since our system deals with vanishing flow when a body part is static by exploiting the lack of flow. Zero flow is bad for pose estimation but good for tracking and we exploit this duality. The 2D predictions are surprisingly good in a range of complex videos. Because no special hardware is required, optical flow may be a useful component in pose estimation, opening up more widespread applications.

While we only use optical flow as input, future work should include additional 2D image cues. Head, feet, and hand detectors could readily be incorporated as, for that matter, depth data from a range sensor or stereo system. Alternatively, FlowCap could be used as a complementary source of information for other pose estimation and tracking methods. For example, we could use FlowCap to initialize more precise model-based trackers. In addition to providing pose, we provide an initial segmentation of the image into regions corresponding to parts. This evidence could readily be incorporated in to existing 2D pose trackers. While our training flow is generated from bodies that are unclothed, we find it generalizes to clothed people. Still, we could simulate sequences of people in clothing (e.g. as in [14]) or use real video of clothed people with ground truth. We could train also FlowCap for specific applications such as TV shows, sports, or video games by constructing training sets with specific motions. Since we start with 3D pose, it would be interesting to directly try to estimate 3D pose, and possibly body shape, from flow (and other cues). Finally our training data could be used to directly train a CNN to estimate pose from flow (and image data). This is an exciting direction that our public dataset[2] will help support.

[2] http://ps.is.tuebingen.mpg.de/project/FlowCap

References

1. http://gpu4vision.icg.tugraz.at.
2. S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *IJCV*, 92(1):1–31, 2011.
3. A. Bissacco, M.-H. Yang, and S. Soatto. Fast human pose estimation using appearance and motion via multi-dimensional boosting regression. *CVPR*, pp. 1–8, 2007.
4. F. Bogo, J. Romero, M. Loper, M. Black: FAUST: Dataset and evaluation for 3D mesh registration. *CVPR*, pp. 3794 –3801. 2014.
5. G.R. Bradski, J.W. Davis: Motion segmentation and pose recognition with motion history gradients. Machine Vision and Applications, 13(3):174–184, 2002.
6. C. Bregler and J. Malik. Tracking people with twists and exponential maps. *CVPR*, pp. 8–15, 1998.
7. A. Efros, A. Berg, G. Mori, and J. Malik. Recognizing action at a distance. *ICCV*, pp. 726–733, 2003.
8. M. Eichner, M. Marin-Jimenez, A. Zisserman, and V. Ferrari. 2D articulated human pose estimation and retrieval in (almost) unconstrained still images. *IJCV*, 99:190–214, 2012.
9. A. Elhayek, E. Aguiar, A. Jain, J. Tompson, L. Pishchulin, M. Andriluka, C. Bregler, B. Schiele, C. Theobalt: Efficient convnet-based marker-less motion capture in general scenes with a low number of cameras. *CVPR,* pp. 3810–3818, 2015.
10. R. Fablet and M. Black. Automatic detection and tracking of human motion with a view-based representation. *ECCV*, pp. 1:476–491, 2002.
11. P. Felzenszwalb, R. Girshick, and D. McAllester. Cascade object detection with deformable part models. *CVPR*, pp. 2241–2248, 2010.
12. V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. *CVPR*, pp. 1–8, 2008.
13. K. Fragkiadaki, H. Hu, and J. Shi. Pose from flow and flow from pose estimation. *CVPR*, pp. 2059–2066, 2013.
14. P. Guan, L. Reiss, D. Hirshberg, A. Weiss, and M. Black. DRAPE: DRessing Any PErson. *SIGGRAPH*, 31(4):35:1–35:10, 2012.
15. D. Hirshberg, M. Loper, E. Rachlin, M. Black: Coregistration: Simultaneous alignment and modeling of articulated 3D shape. *ECCV,* pp. 242–255. LNCS 7577, IV, Springer, 2012.
16. A. Jain, J. Tompson, Y. LeCun, C. Bregler: MoDeep: A deep learning framework using motion features for human pose estimation. *ACCV*. pp. 302–315, 2014.
17. S. Ju, M. Black, and Y. Yacoob. Cardboard people: A parameterized model of articulated motion. *Face and Gesture*, pp. 38–44, 1996.
18. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *JMLR*, 12:2825–2830, 2011.
19. B. Sapp, D. Weiss, and B. Taskar. Parsing human motion with stretchable models. *CVPR*, pp. 1281–1288, 2011.
20. L. Schwarz, A. Mkhitaryan, D. Mateus, and N. Navab. Estimating human 3D pose from time-of-flight images based on geodesic distances and optical flow. *Face and Gesture*, pp. 700–706, 2011.
21. J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, and A. Blake. Efficient human pose estimation from single depth images. *PAMI*, 35(12):2821–2840, Dec. 2013.
22. H. Sidenbladh, M. Black, and D. Fleet. Stochastic tracking of 3D human figures using 2D image motion. *ECCV*, pp. 702–718, 2000.

23. L. Sigal, A. Balan, and M. Black. HumanEva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *IJCV*, 87(1):4–27, Mar. 2010.
24. S. Wachter and H. Nagel. Tracking persons in monocular image sequences. *CVIU*, 74(3):174–192, 1999.
25. G. Welch and G. Bishop. An introduction to the Kalman filter. UNC, *TR 95-041*, 2006.
26. M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof. Anisotropic Huber-L1 optical flow. *BMVC*, pp. 108.1–108.11, 2009.
27. Y. Yang and D. Ramanan. Articulated pose estimation using flexible mixtures of parts. *CVPR*, pp. 1385–1392, 2011.
28. H. Zou, T. Hastie: Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society. Series B (Statistical Methodology) 67(2), pp. 301–320, 2005
29. S. Zuffi, J. Romero, C. Schmid, and M. Black. Estimating human pose with flowing puppets. *ICCV*, pp. 3312–3319, 2013.