POOF: Efficient Goalie Pose Annotation using Optical Flow

Brennan Gebotys, Alexander Wong and David Clausi
Systems Design Engineering, University of Waterloo, Waterloo, Canada

Keywords: Computer Vision, Neural Networks, Pose Estimation, Optical Flow, Annotation.

Abstract: Due to the wide range of applications for human pose estimation including sports analytics and more, research has optimized pose estimation models to achieve high accuracies when trained on large human pose datasets. However, applying these learned models to datasets that are from a different domain (which is usually the goal for many real-world applications) usually leads to a large decrease in accuracy which is not acceptable. To achieve acceptable results, a large number of annotations is still required which can be very expensive. In this research, we leverage the fact that many pose estimation datasets are derived from individual frames of a video and use this information to develop and implement an efficient pose annotation method. Our method uses the temporal motion between frames of a video to propagate ground truth keypoints across neighbouring frames to generate more annotations to provide efficient POse annotation using Optical Flow (POOF). We find POOF achieves the best performance when used in different domains than the pretrained domain. We show that in the case of a real-world hockey dataset, using POOF can achieve 75% accuracy (a +15% improvement, compared to using COCO-pretrained weights) with a very small number of ground truth annotations.

1 INTRODUCTION

There are numerous applications for human pose estimation including sports analytics, sign language recognition and more. Specifically, in sports analytics for hockey, extracting accurate poses can lead to greater insight into players performance including the player's form (e.g, skating with or without the correct form), classification of specific actions (e.g, slapshot, wrist shot, etc.), and the quantification of the probability of scoring (or probability of saving a shot, in the goalie's case). These insights can then be used to improve the team or help develop a plan against an upcoming opponent.

Because of the wide range of applications, pose estimation models have been researched extensively (Li et al., 2019; Newell et al., 2016; Lin et al., 2014). This research has resulted in most pose estimation models achieving high accuracies when they are trained on large datasets. However, when transferring these learned models to other visually different datasets (which is the case for many real-world applications), usually, the model accuracy significantly decreases and leads to unsatisfactory results. To obtain satisfactory results, large amounts of labelled pose data from the new dataset are still required. However, for some labs/companies, this is not possible due to resource and time constraints. This leads to the problem of performing sufficiently accurate pose estimation with only a small number of annotations.

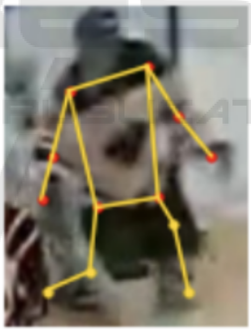

Figure 1: An example of a predicted goalie pose from the NHL goalie dataset used throughout the experiments.

A common theme of human pose datasets is that each example is a frame that has been extracted and annotated from a corresponding video. In this research, we leverage the inherent motion found in videos and use optical flow estimation between frames to propagate annotations from one frame to its neighbouring frames. Doing so results in a multiplicative increase in pose annotations with no additional cost. We call this method, POOF (POse annotation using Optical Flow).

Gebotys, B., Wong, A. and Clausi, D.
POOF: Efficient Goalie Pose Annotation using Optical Flow.
DOI: 10.5220/0010657000003059
In *Proceedings of the 9th International Conference on Sport Sciences Research and Technology Support (icSPORTS 2021)*, pages 116-122
ISBN: 978-989-758-539-5; ISSN: 2184-3201

To investigate POOFs performance, we run extensive experimental studies on an NHL (National Hockey League) goalie-pose dataset which contains many similar features to real-world datasets including but not limited to: dataset-specific poses which are not common in the large datasets, joint occlusions caused by hockey players skating in front of the camera, and image blurriness caused from camera movement. Also, to further investigate how POOF generalizes in other settings, we perform ablation studies across a variety of pretrained weights and hyperparameters.

2 RELATED WORK

In this section, we describe research that investigates how to perform pose estimation with a small number of examples and how the research relates to POOF. The research solutions can be generally described as either improving the annotation generation (similar to POOF) or modifying the model directly.

(Neverova et al., 2019) used motion to extend keypoints to neighbouring frames for dense keypoint estimation. We extend this work by applying it to pose estimation and investigate the performance on out-of-domain and smaller datasets where POOF is determined to perform effectively. Furthermore, we show POOF can lead to improved performance up to a radius of 10 frames whereas (Neverova et al., 2019) only investigated using a radius of 3 frames. We find this results in more than triple the number of labels and further improved performance.

Rather than optimize the annotations, (Bertasius et al., 2019) used a semi-supervised approach to learn a model from sparse video annotations. However, their approach requires annotations between every n-th frame in a video (in their paper they used every 7-th frame), which our approach doesn't require. By removing this requirement our method significantly reduces the number of annotations required.

(Romero et al., 2015) showed that it's possible to predict keypoints using only optical flow and Kalman filters, without any ground truth labels. We further extend this research by incorporating a small number of annotations that we believe are easy to collect. Instead of using motion information, (Charles et al., 2016) used visual features to propagate keypoints across neighbouring frames.

Pose estimation and optical flow have also been shown to be very complementary. (Pfister et al., 2015) developed a model which takes multiple frames and optical flow estimation as input and showed an improvement in pose estimation accuracy. (Zhang et al., 2018) used pose estimation to improve the representation of motion estimation for humans.

Rather than propagate keypoints, another way to generate more annotations is to use synthetically generate data. (Doersch and Zisserman, 2019) found that pasting generated humans in specific augmented poses across a variety of background images can lead to improved generalization performance for 3D pose estimation. (Hinterstoisser et al., 2019) used a similar approach for object detection and found improved performance. However, these techniques usually require additional data to get working (e.g, segmentation information of the poses to be able to paste on different backgrounds) which can be costly.

3 METHODOLOGY

Before describing the methodology we first define a few terms. We define a hyperparameter, R, as the number of frames before and after the ground truth annotation to which the keypoints will be propagated. We define K_t as a vector of x-y coordinates representing keypoints in the t-th frame.

We define $M_{i,j}$ as the optical flow estimation between the i-th and j-th frame represented as a $A \times B \times 2$ matrix where $A \times B$ is the size of each frame. The coordinates (k,l) are referenced in $M_{i,j}$ using $M_{i,j,(k,l)}$, which represents how the pixels of the i-th frame at coordinates (k,l) moved to the j-th frame in terms of a change in the x and y coordinates.

The first step of our method requires collecting ground truth annotations across a video. We aim to have diverse annotations which cover a variety of poses that are temporally far apart from each other. Ideally, we want to select annotations that are at least $2 \times R$ frames apart. This is because when we propagate the ground truth keypoints to the nearest R frames, if the ground truth keypoints frames are $2 \times R$ apart, there will be no overlap in predictions and we will maximize the amount of annotated data created.

For each ground truth annotation at time t, K_t, we use an optical flow estimation model to predict the motion between consecutive frames to obtain $M_{t,t+1}$ $\forall t \in [t-R, t+R-1]$.

We then predict the keypoints which surround the ground truth annotation frame, K_{t-1} and K_{t+1}, using the annotated keypoints K_t and the motion between the frames, $M_{t-1,t}$and $M_{t,t+1}$, as follows:

$$K_{t-1} = K_t - M_{t-1,t,K_t} \tag{1}$$

$$K_{t+1} = K_t + M_{t,t+1,K_t} \tag{2}$$

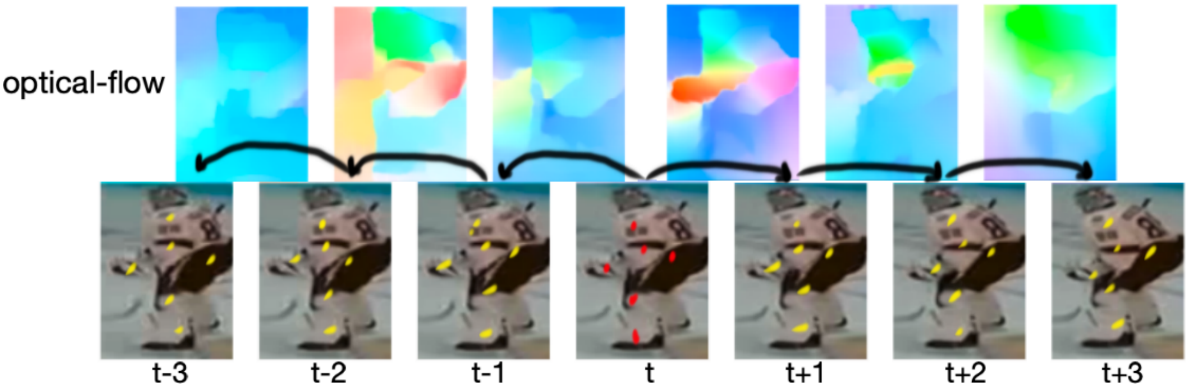

Figure 2: Visual description of the POOF method for $R = 3$. The annotated keypoints (shown in red) are propogated to annotate the surrounding frame's keypoints (shown in gold) using the optical flow between consecutive frames (images in the top row).

where $M_{t,t+1,K_t}$ is $M_{t,t+1}$ indexed at the coordinates of K_t.

We repeat equation 1 $\forall t \in [t-R,t)$ and equation 2 $\forall t \in (t,t+R]$ to obtain keypoints $\forall t \in [t-R,t+R]$.

Figure 2 shows a visual description of POOF for $R = 3$. First, the optical flow is computed between consecutive frames (images in the top row). The ground truth annotated keypoint at time t (shown in red) is then propagated to annotate the keypoints for its neighbouring frames (shown in gold).

4 EXPERIMENTS

In the following section, we describe and report on experiments using POOF. Specifically, we first define the specific models and datasets used and then perform multiple ablation experiments to understand settings where POOF performs the best.

4.1 Setup

Throughout the experiments, we used the publicly-available code for MSPN (Li et al., 2019) and RAFT (Teed and Deng, 2020) as pose and optical flow estimation models respectively. We trained our pose estimation model for 10 epochs with a learning rate of 0.01 and a batch size of 32. For the optical flow estimation model, we used the publicly-available pretrained weights from the Sintel dataset (Butler et al., 2012).

4.2 Metrics

We also record the validation accuracy of the model and refer to it as "Accuracy" in the experiment tables. We define a keypoint to be accurate if the mean absolute error (MAE) between the predicted keypoint and the ground truth keypoint is less than a threshold of 20 pixels. We chose a threshold of 20 through visual inspection of different MAE distances on different examples. We also perform experiments on different threshold values in Section 4.7.

4.3 Datasets

We perform our experiments on an NHL video broadcast dataset. This dataset was selected because it includes common real-world pose estimation challenges such as dataset-specific poses which are not common in large datasets, joint occlusions caused by hockey players skating in front of the camera, and image blurriness caused by camera movement. Furthermore, the visual appearance of an NHL game is much different compared to the images in larger benchmarks such as COCO (Lin et al., 2014), which again, is the case for most real-world datasets.

Throughout the data, the hockey goalie has been cropped out of the broadcast video and resized to a 256 x 192 image. The ground truth training examples were manually selected to be sparse, non-uniform, and contain a variety of poses across 6 different broadcast videos. The same approach was used for the validation examples, but across 2 broadcast videos (not included in the training set) and resulted in 16 total labels. Throughout the experiments, we used a radius of 10 (R=10) unless stated otherwise.

Figure 1 shows an example from the dataset as well as the predicted pose from a model which has been pretrained on the large pose estimation dataset, COCO (Lin et al., 2014). We can see that the model incorrectly classified the goalie's pose. This is likely because goalie images are visually different from ex-

Table 1: Accuracy of different pose-estimation data for hockey goalie pose estimation.

Init Weights	Training Data	# Examples	Accuracy
COCO	None	0	60.53
	GT Labels	69	23.03
	POOF	69 + 1314	**75.66**

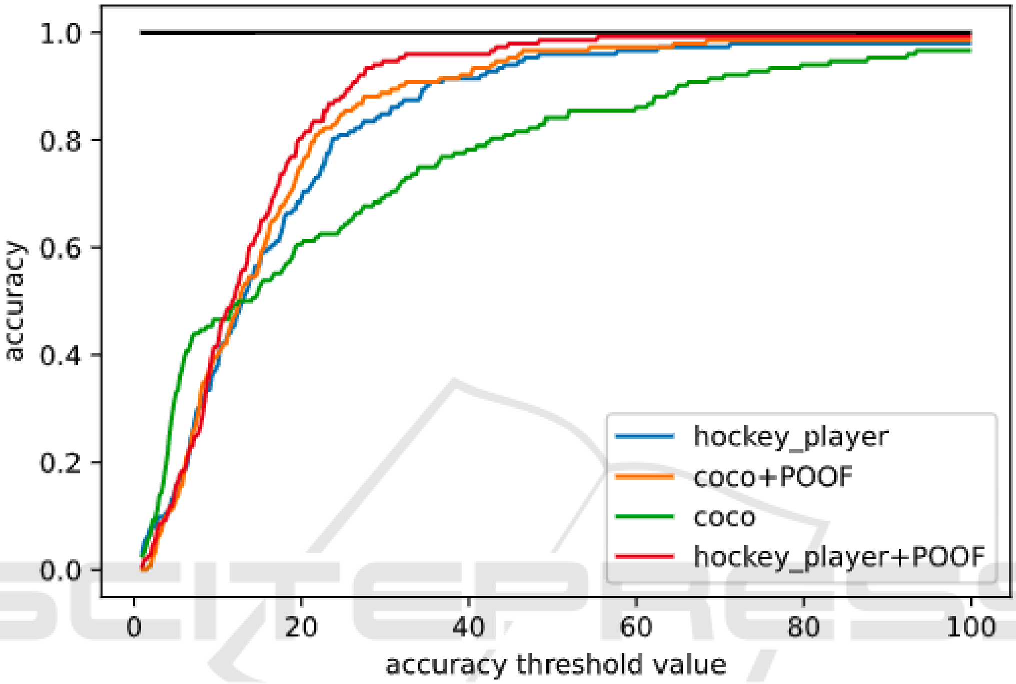

Figure 3: Diagram with the accuracy on the y-axis and the accuracy threshold value on the x-axis. We compared results using hockey player pretrained weights (hockey_player), COCO pretrained weights (COCO) both with (+POOF) and without POOF.

amples in the COCO dataset and the goalie is in a dataset-specific pose (e.g, the hockey goalie is on his knees).

4.4 POOF Ablation Study

Table 1 compares accuracies across different types of training data using pretrained weights from the COCO dataset. We compared three types of training data: 'None' evaluates the pretrained model directly on the dataset, 'GT Labels' finetunes the model on the manually-annotated/ground-truth (GT) data, and our proposed method 'POOF' trains the model on the manually annotated data as well as the optical flow propagated data.

In the second row of Table 1, we see that using a small number of ground truth labels (69) leads to a decrease in accuracy of -37% compared to using only pretrained weights (from 60% to 23%). This shows that using a small number of labels is worse than using no labels.

We also see in the third row, using POOF, achieves an increase of 15% accuracy (from 60% to 75%) over using only pretrained weights while using the same number of ground truth annotations as GT Labels. This shows that POOF can significantly improve the performance compared to using only pretrained weights with only a small number of annotations.

4.5 The Effects of Pretrained Weights

We also perform experiments to understand the effect of using different pretrained weights. Table 2 shows the results of using randomly initialized weights (None), pretrained weights from the COCO dataset (COCO), and pretrained weights from a very similar hockey player dataset which includes the hockey players instead of the hockey goalies (Hockey Players).

We see in Table 2, from COCO pretrained weights, POOF can significantly outperform the GT Labels and pretrained weights. However, when using

Table 2: Accuracy of different pose-estimation data for hockey goalie pose estimation.

Init Weights	Training Data	Accuracy
None	None	0.00
	GT Labels	0.06
	POOF	**38.82**
COCO	None	60.53
	GT Labels	23.03
	POOF	**75.66**
Hockey Players	None	69.08
	GT Labels	**80.92**
	POOF	80.26

pretrained weights from the hockey player dataset, we see POOF leads to about the same performance. From these results, we hypothesize that when the domains of the pretrained weights and the new dataset are similar, the performance improvement from POOF is minimal, however, POOF excels when the domains between the pretrained weights and the new dataset are different. The minimal improvement finding agrees with the results by (Neverova et al., 2019).

When not using pretrained weights (None), POOF significantly outperforms training on ground truth labels and increases the accuracy by 38% (from 0.06 to 38.82). This is very valuable in the case of annotating new keypoints which are not included in large benchmarks. Specifically, since keypoints that are not included in the large pretraining benchmarks (e.g, hockey stick keypoints, corner of goalie pads, etc.) would not have any pretraining data, they would have to use randomly initialized weights (None). Table 2 shows that POOF achieves a significant accuracy improvement for these new keypoints. This result was not discovered in previous research.

4.6 Propagation Radius

Table 4 shows the accuracy achieved using different values of R (which defines the number of neighbouring frames). Note that the dataset used inverted keypoints to COCO (the left shoulder of COCO is the right shoulder in this data, etc.) and so the accuracy results are different from previous experiments.

Table 4 shows that the best accuracy is achieved when $R = 10$ frames. We hypothesize that using $R = 5$ resulted in lower performance due to having too few labels. And we hypothesize that using a $R = 20$ results in too many incorrect annotations due to occlusions (e.g, hockey players skating between the camera and goalie), blurriness (e.g, from camera movement), and small errors in close frames which result in larger errors in frames further away.

We recommend, that in practice, R should be selected based on the data. If occlusions and blurriness are minimized throughout the dataset then keypoint propagation should work better for a longer distance and so a larger R value should be chosen. However, if occlusions and blurriness occur often in the dataset, then a lower R value should be chosen to reduce the number of incorrect annotations.

4.7 Various Accuracy Thresholds

To further investigate the performance improvement of POOF, we also investigated the results across different accuracy thresholds.

Figure 3 shows the accuracy (shown on the y-axis) across different accuracy thresholds (shown on the x-axis) which represents the maximum distance a keypoint can be from the ground truth and still be classified as correct. The black line is a straight line that represents 100% accuracy. The steepness of the slope is indicative of a better model.

We can see that the lines which use POOF (orange and red) are much steeper than the lines which don't (green and blue). Specifically, if we look at the orange vs green and red vs blue lines, we see that the improvement using POOF is significant across many accuracy thresholds. This further confirms POOF improves model performance.

4.8 Change in Per-joint Accuracy

Lastly, to further understand where the performance improvement is coming from, we investigated the accuracy improvement of each joint when using POOF.

Table 3 shows the accuracy across all the joints. The joint names are formatted to have the side of the body, followed by an underscore, followed by the body part (e.g, the left shoulder keypoint is formatted as L_shoulder). The second column shows the results of the initial weights used (without any training) (e.g, COCO) and the third column (e.g, +POOF) shows the results after applying POOF. Also, the same format is in the fourth and fifth columns which are used to compare using pretrained weights from the hockey player dataset without POOF (e.g, HockeyPlayer) and with POOF (e.g, +POOF). We show the percentage improvement achieved when using POOF in brackets.

We see that POOF consistently improves the accuracy of most joints by a significant amount (e.g, +40% L_wrist in the COCO row). However, POOF also sometimes results in poorer accuracy (e.g, -9% R_ankle in the HockeyPlayer row). We hypothesize this could be due to the model overfitting the noise

Table 3: Accuracy on specific joints with (+POOF) and without POOF using different pretrained weights (e.g, COCO and HockeyPlayer). Change in accuracy using POOF in brackets.

Joint	COCO	+POOF	HockeyPlayer	+POOF
L shoulder	86	93 (+7)	80	93 (+13)
R shoulder	100	100 (+0)	100	100 (+0)
L elbow	50	64 (+14)	71	78 (+7)
R elbow	80	80 (+0)	90	90 (+0)
L wrist	26	66 (+40)	40	53 (+13)
R wrist	58	50 (-8)	33	58 (+25)
L hip	58	83 (+25)	66	91 (+25)
R hip	81	72 (-9)	63	72 (+9)
L knee	57	85 (+27)	57	85 (+28)
R knee	33	50 (+17)	66	75 (+9)
L ankle	66	80 (+14)	80	86 (+6)
R ankle	41	75 (+34)	91	83 (-8)
Mean	61	74 (+13)	69	80 (+11)

Table 4: Accuracy using different radius sizes while propagating the labels with POOF.

R	# Examples	Accuracy
5	255	51.64
10	420	**61.50**
20	670	35.21

in the propagated keypoints. In practice, this could be solved by using an ensemble of models where for each keypoint the best performing model is used to predict it.

5 FUTURE RESEARCH

In this section, we describe some limitations of POOF and potential future research directions.

One limitation of our research is that we only tested POOF on a single hockey goalie dataset. It would be interesting to experiment across a wider variety of datasets and to assess consistency across other datasets. Specifically, it would be interesting to see if the results held across different sports such as soccer or basketball where the person's motion and the video characteristics are very different compared to hockey.

Another avenue for future research is to further investigate the effect of using different *R* values. In our research, we only investigated three potential values, but it would be interesting to test more values to further understand their relationship to performance. As well, ideally, we would want to reduce the importance of selecting the correct hyperparameter so one could investigate how to select *R* quantitatively rather than qualitatively.

One of the main limitations with POOF is that the optical flow estimation is unable to account for keypoints that start as visible and later become occluded by either another object occluding the keypoints or through the person rotating in a way that occludes the keypoint. Furthermore, POOF is also unable to account for keypoints that were labelled as occluded but become visible later in the video. Different solutions could be experimented with to solve this problem which could allow us to label longer sequences. This would further increase the number of annotations while also reducing the amount of noise in the propagated annotations. This would be likely to lead to further improvement in model performance. One potential solution could be to incorporate visual information in the keypoint propagation stage similar to (Charles et al., 2016).

6 CONCLUSION

In this paper, we introduced POOF, a data-efficient pose annotation method that utilizes optical flow to propagate ground truth annotations to neighbouring frames. POOF improves on the previous work of pose estimation solutions by removing data annotation constraints such as requiring a ground truth keypoint every n-frames and shows it performs best when transferring models between different domains (in Table 2). Using a hockey goalie dataset, we show that POOF can improve performance with a very small amount of labels. We also show POOF can achieve significantly improved results over using pretrained weights across various accuracy thresholds. Furthermore, we showed this performance improvement is achieved across most individual joints and also suggested multiple directions for future research. Overall, this research should significantly reduce the time required for annotating pose data across different domains without compromising model accuracy and allow pose estimation to be more easily applied to a wide variety of domains.

ACKNOWLEDGEMENTS

This research is supported in part by grants from MITACs and Stathletes, Inc.

REFERENCES

Bertasius, G., Feichtenhofer, C., Tran, D., Shi, J., and Torresani, L. (2019). Learning temporal pose estimation from sparsely-labeled videos. *arXiv*, (NeurIPS):1–12.

Butler, D. J., Wulff, J., Stanley, G. B., and Black, M. J. (2012). A naturalistic open source movie for optical flow evaluation. In *European Conference on Computer Vision*, pages 611–625. Springer.

Charles, J., Pfister, T., Magee, D., Hogg, D., and Zisserman, A. (2016). Personalizing human video pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3063–3072.

Doersch, C. and Zisserman, A. (2019). Sim2real transfer learning for 3d human pose estimation: motion to the rescue. *arXiv preprint arXiv:1907.02499*.

Hinterstoisser, S., Pauly, O., Heibel, H., Marek, M., and Bokeloh, M. (2019). An annotation saved is an annotation earned: Using fully synthetic training for object instance detection. *arXiv preprint arXiv:1902.09967*.

Li, W., Wang, Z., Yin, B., Peng, Q., Du, Y., Xiao, T., Yu, G., Lu, H., Wei, Y., and Sun, J. (2019). Rethinking on multi-stage networks for human pose estimation. *arXiv preprint arXiv:1901.00148*.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer.

Neverova, N., Thewlis, J., Guler, R. A., Kokkinos, I., and Vedaldi, A. (2019). Slim densepose: Thrifty learning from sparse annotations and motion cues. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10915–10923.

Newell, A., Yang, K., and Deng, J. (2016). Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, pages 483–499. Springer.

Pfister, T., Charles, J., and Zisserman, A. (2015). Flowing convnets for human pose estimation in videos. In *Proceedings of the IEEE international Conference on Computer Vision*, pages 1913–1921.

Romero, J., Loper, M., and Black, M. J. (2015). Flowcap: 2d human pose from optical flow. In *German Conference on Pattern Recognition*, pages 412–423. Springer.

Teed, Z. and Deng, J. (2020). Raft: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision*, pages 402–419. Springer.

Zhang, D., Guo, G., Huang, D., and Han, J. (2018). Poseflow: A deep motion representation for understanding human behaviors in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6762–6770.