Fast Human Pose Estimation in Compressed Videos

Huan Liu, *Graduate Student Member, IEEE*, Wentao Liu, Zhixiang Chi, Yang Wang, Yuanhao Yu, Jun Chen, *Senior Member, IEEE*, and Jin Tang

***Abstract*—Current approaches for human pose estimation in videos can be categorized into per-frame and warping-based methods. Both approaches have their pros and cons. For example, per-frame methods are generally more accurate, but they are often slow. Warping-based approaches are more efficient, but the performance is usually not good. To bridge the gap, in this paper, we propose a novel fast framework for human pose estimation to meet the real-time inference with controllable accuracy degradation in compressed video domain. Our approach takes advantage of the motion representation (called "motion vector") that is readily available in a compressed video. Pose joints in a frame are obtained by directly warping the pose joints from the previous frame using the motion vectors. We also propose modules to correct possible errors introduced by the pose warping when needed. Extensive experimental results demonstrate the effectiveness of our proposed framework for accelerating the speed of top-down human pose estimation in videos.**

***Index Terms*—Human pose estimation, compressed video, deep neural network.**

I. Introduction

HUMAN pose estimation in videos is a cornerstone for many computer vision applications, such as smart video surveillance, human-computer interaction, virtual reality *etc*. It aims to seek for locations of human body joints (*e.g*. head, elbow and *etc*.) in video sequences. Current real-time solutions to this problem can be categorized into per-frame methods [7], [10], [11], [13], [19], [22], [24], [30], [32]–[35], [40], [46]–[48], [50], [53], [56] and warping-based methods [5], [14], [38], [43].

Due to their simplicity, per-frame methods are widely deployed in real-world applications. In general, the per-frame methods can be categorized into top-down methods [32], [46], [53], and bottom-up methods [7], [13]. While bottom-up methods localize human joints for all persons in a frame, top-down methods decompose the multi-person human pose estimation into a simpler task of single-person pose estimation by first detecting each person in a frame, then applying a single-person pose estimation on each detected person. Although the two different pipelines have their distinctive properties, both of them are usually designed to meet the real-time demand from the perspective of searching compact neural network models or reducing the input image size. However, per-frame methods do not consider the temporal continuity between frames. As a result, they involve a lot of redundant computations.

To exploit temporal continuity in videos, warping-based methods aim to discover temporal relations (*e.g*. optical flow [38], [43], pose flow [55], *etc*.) and quickly propagate human pose from one frame to another. However, computing optical flow is often time-consuming, so warping-based methods are rarely used in real-world applications.

In this paper, we introduce an alternative way of exploiting the temporal continuity in videos for human pose estimation. The core idea of our approach is to take advantage of the motion information that is already available in compressed videos when they are being encoded by standard video codecs. Compressed video streams only retain very few frames as RGB images, but contain massive motion information (*i.e*. motion vector and residual error) for frame reconstruction. These motion vectors and residual error are readily available in compressed videos and do not require any computation to obtain. Recent years have witnessed many successes in handling computer vision tasks in the compressed video domain. Some early work focuses on classification tasks such as action recognition [52], video classification [8], [9]. These tasks usually do not require precise motion cues at the pixel level, so motion vectors in compressed videos can be easily applied. There are also works on semantic segmentation [16], [29] in compressed videos. Although semantic segmentation is a pixel labeling task, the performance of semantic segmentation is largely influenced by the prediction in the interior of object instances rather than instance boundaries. As a result, this task does not require very much motion information either. In comparison, human pose estimation in compressed videos is much more challenging, since this task requires accurate joint predictions.

To this end, we propose a novel framework for human pose estimation in the compressed video domain. The framework consists of four components, *i.e*. human pose estimator, fast pose warping module (FPW), pose recall module (PR) and transition re-initialization module (TR). To be specific, the human pose estimator is a top-down pose estimation network working on RGB images. For the purpose of reducing temporal redundancy, a fast pose warping module is designed to use motion vectors

Manuscript received 20 July 2021; revised 26 November 2021; accepted 27 December 2021. Date of publication 11 January 2022; date of current version 12 April 2023. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ramanathan Subramanian. *(Corresponding author: Huan Liu.)*

Huan Liu and Jun Chen are with the Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON L8S 2K1, Canada (e-mail: liuh127@mcmaster.ca; chenjun@mcmaster.ca).

Wentao Liu, Zhixiang Chi, Yuanhao Yu, and Jin Tang are with the Noah's Ark Laboratory, Huawei Technologies Canada, Markham, ON L3R 5A4, Canada (e-mail: wentao.liu2@huawei.com; zhixiang.chi@huawei.com; yuanhao.yu@huawei.com; tangjin@huawei.com).

Yang Wang is with the Department of Computer Science, University of Manitoba, Winnipeg MB R3T 2N2, Canada and Huawei Technologies Canada, Markhan, ON L3R 5A4, Canada (e-mail: ywang@cs.umanitoba.ca).

Digital Object Identifier 10.1109/TMM.2022.3141888

for rapid pose propagation across consecutive frames. However, since motion vectors are noisy and not always associated with the motion on the body parts, we design a pose recall module to adaptively find "hard-to-warp" human instances and perform human pose estimation instead of warping by jointly considering the motion intensity and confidence on body joints. Moreover, video transitions can result in significant motion cues which are irrelevant to body motion. To address this issue, the transition re-initialization module is introduced to terminate the warping process at video transitions and switch to RGB-based pose estimation.

The main contributions of this work can be summarized as follows. First, this paper represents the first work on real-time human pose estimation in the compressed domain. Second, we propose a human pose estimation framework in the compressed domain using three well-designed modules. Finally, we demonstrate through extensive experimental results that our framework can speed up existing per-frame and warping-based methods by 2-5 times on the Posetrack dataset, while achieving comparable performance in accuracy.

II. Related Works

In this section, we briefly review several lines of research related to our work.

Per-frame Human Pose Estimation: Traditional human pose estimation methods [2], [3], [23], [39] usually adopt the pictorial structures model with hand-crafted features. These methods often fail when some body parts are occluded. In recent years, with the emerging of deep convolutional neural networks, most of the image-based human pose estimation [7], [10], [11], [13], [19], [22], [24], [25], [30], [32]–[35], [40], [46]–[48], [50], [53], [56], [58] learn to predict human poses on large-scale datasets with intensive human joints annotations. Instead of mapping images directly to human joint coordinates, most of these methods, except for [48], choose to predict heatmaps for easier regression and optimization.

In the era of deep learning, image-based methods can be categorized as top-down methods and bottom-up methods. Top-down methods [11], [19], [32], [35], [46], [53] usually rely on a human detector that helps localizes human instances in an image. Then the methods decompose the multi-person human pose estimation task into single person pose estimation problems. On the contrary, bottom-up methods [7], [13], [22] first detect all the body joints in an image, and then assign the detected joints to each person.

These works mainly focus on exploring novel models to achieve state-of-the-art human pose estimation accuracy, but their processing speed is often slow.

Fast Human Pose Estimation: Although the efficient estimation of the human pose is quite important, very few works aim for this goal. Rafi *et al.* [41] introduce a compact neural network that can be trained efficiently on a mid-range GPU. Bulat *et al.* [6] binarize heavy CNN architectures for model compression and specifically designed a parallel and multi-scale architecture for the binary case. Zhang *et al.* [56] successfully employ a well-trained large network to help boost the performance of a small network with knowledge distillation [20]. However, the above-listed methods only focus on designing a small network that is cost-effective for deploying in practice. In this paper, we alternatively investigate the possibility of accelerating inference speed in the video compressed domain.

Video Based Human Pose Estimation: Temporal dependency among video frames is the most crucial factor that distinguishes an image task from a video task. Exploiting the temporal correlation wisely can significantly improve the performance in a video task. However, due to the scarcity of large-scale video-based benchmarks, video-based human pose estimation has only drawn very little attention in recent years. Some methods [38], [43] use dense optical flow as temporal representations to capture relationships across the multiple frames. In contrast, Doering *et al.* [14] compute task-specific motion representation only on human joints to reduce redundancy of dense optical flow. Bertasius *et al.* [5] introduce a novel CNN architecture for pose estimation in sparsely labeled videos. This method uses a neural network to directly learn offsets of consecutive frames. Although most of these video-based methods show great improvements on estimation accuracy, they still ignore the problem of how to efficiently estimating human pose in videos.

Video Analysis in Compressed Domain: Video analysis in the compressed domain is also understudied. There are few works that try to leverage the compressed domain knowledge to assist specific video analysis tasks. The current methods in compressed domain can be categorized as traditional methods and deep learning base methods. For traditional methods, Chen *et al.* [12] propose to use global motion estimation and Markov random field for extraction moving regions in compressed domain. Some works [28], [37] introduce fast scene change detection algorithm using the feature from compressed videos. The two methods mainly focus on how to precisely detect wipe transition. Despite the effectiveness of traditional method, they usually adopt compressed knowledge for transition and motion detection rather than high-level video analysis. To further exploit the valuable information in compressed domain, some recent work proposes to use deep learning techniques for video analysis in the compressed video domain. There is some work [8], [9] on 3D convolutional neural networks for video classification utilizing compressed domain knowledge. Wu *et al.* [52] accelerate action recognition directly on compressed videos. The success of extracting high-level representations from the compressed domain implies the potential of compressed domain information in other computational vision tasks. Recently, Li *et al.* [29] adopt convolutional LSTM to propagate semantic maps to consecutive frames by motion vector and residual. Feng *et al.* [16] propose a novel real-time framework for semantic segmentation using compressed domain knowledge. Due to the nature of semantic segmentation, where most of the pixels are inside of objects, the noise in motion vectors can be largely tolerated. On the contrary, accurately propagating the human joints with noisy motion vectors is a more challenging task. In this paper, we are inspired by Feng *et al.* [16] to propose a method for fast human pose estimation in the compressed domain. To our best knowledge, this paper is the first to address this problem.

Video Analysis Beyond RGB Frames: Our work is loosely related to other video analysis tasks that use the information beyond RGB frames in a video. For example, there has been lots of work (e.g. [26], [42], [57]) on using depth information in RGBD videos for object recognition, pose estimation, etc. However, these works can only work on video data collected by RGBD cameras, since the depth information is not available in regular videos. In contrast, our work is more widely applicable since the motion vector information is readily available in any compressed video.

III. Background: Compressed Video

Due to the enormous data volume, digital videos are typically encoded into video streams for efficient storage and transmission. Commonly used modern video codecs include MPEG-4 Part 2 [27], H.264/AVC [51], HEVC [44], VP9 [31], etc. A video stream compressed by these video codecs has a very different structure from a sequence of stand-alone images as often seen in an uncompressed video. In this section, we take the MPEG-4 Part 2 (Simple Profile) codec [27] as an example to analyze the type of data that are available in a video stream. Nevertheless, most popular video codecs share a similar predictive coding strategy and generate compressed streams with a similar structure. So our analysis on this particular codec generalizes to other codecs.

The basic unit in a compressed video is called a group of pictures (GOP). The encoding and decoding processes of one GOP are independent of any other GOPs. A compressed video is composed of a sequence of such GOPs. In the default mode of the MPEG-4 Part 2 codec, a GOP consists of 12 frames, with the first being an I-frame (intra-coded frame) and the rest being P-frames (predictive frames). Video codecs treat the two types of frames differently. The I-frame is encoded as a regular image, so decoding it does not depend on any other frames in the GOP. However, the encoding of each P frame depends on the data from its previous frame, which finally relies on the data of the first I-frame of the GOP. Specifically, for each 16×16 block in a P-frame $\mathbf{I}^t$ at time t, the codec first tries to find a best-matched block in the previous frame $\mathbf{I}^{t-1}$ by a block-matching method [4]. It then represents the correspondence between the two blocks by a vector pointing from the reference block to the target. Such a vector is known as the motion vector (MV) in the context of video compression. After the block matching, the residual between the target and reference blocks is also computed and encoded into the video stream. As such, the P-frame $\mathbf{I}^t$ is compactly represented by an MV map $\mathbf{M}^t$ and a residual map $\mathbf{R}^t$, and can be reconstructed by reusing the data of the previous frame $\mathbf{I}^{t-1}$.

$$\mathbf{I}^t(x, y) = \mathbf{I}^{t-1}\left[(x, y) - \mathbf{M}^t(x, y)\right] + \mathbf{R}^t(x, y), \qquad (1)$$

where (x, y) indicates any pixel position in the frame. Fig. 1 illustrates the representation and the reconstruction process of P-frames in a GOP. Some other codecs may generate another type of frame, *i.e.* B-frame (bi-directional frame), which is encoded in a similar manner to P-frames except that the motion vectors are estimated from both previous and future frames.

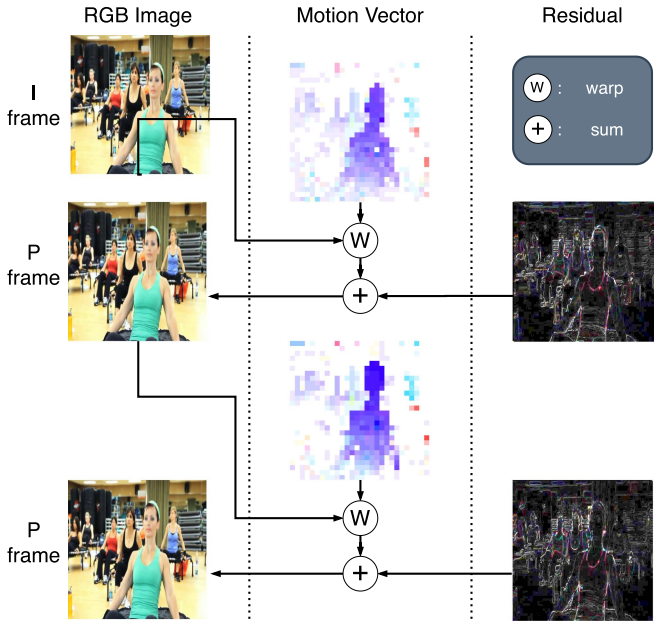

Fig. 1. Illustration of decoding a compressed video. Each I frame is encoded as a regular image. Each P frame is stored as a motion vector and residual that represent the correlation between the current P frame and the previous frame.

IV. Our Approach

Top-down pose estimation is often performed in a two-stage manner. First, a human detector scans the whole image to crop out each person instance in a bounding box. Then, pose estimation is performed in each of the bounding boxes to localize each joint of the person using a heatmap. This process is well established for pose estimation on a still image but still has room to improve when processing a video. As analyzed in Section III, neighboring frames are highly correlated with each other, so it is intuitively possible to reuse the estimation results from the previous frame in the current frame. Let us consider an extreme case where a person is doing yoga and keeping a posture for a few seconds. The motion vectors in the video will indicate that there is no motion between adjacent frames. We can then perform pose estimation only on the first frame and copy the results to the remaining frames. Intuitively, this approach can save several folds of inference time while achieving a similar level of accuracy.

Our proposed approach is inspired by and a natural extension of this intuition. By exploiting the inter-frame relationship readily available in a compressed video stream, we design a system that can accelerate any per-frame pose estimation method while maintaining relatively high prediction accuracy. As shown in Fig. 2, our proposed approach contains several components: a human instance detector, a single-person pose estimator, a fast pose warping module (FPW), a pose recall (PR) module and a transition re-initialization module (TR). In particular, the first two components form the baseline image-based pose estimator that is used to initialize the human pose in I-frames. The last three modules are designed for accelerating and correcting pose estimation in P-frames. First, we design an FPW module to propagate the joints of each person based on the results in

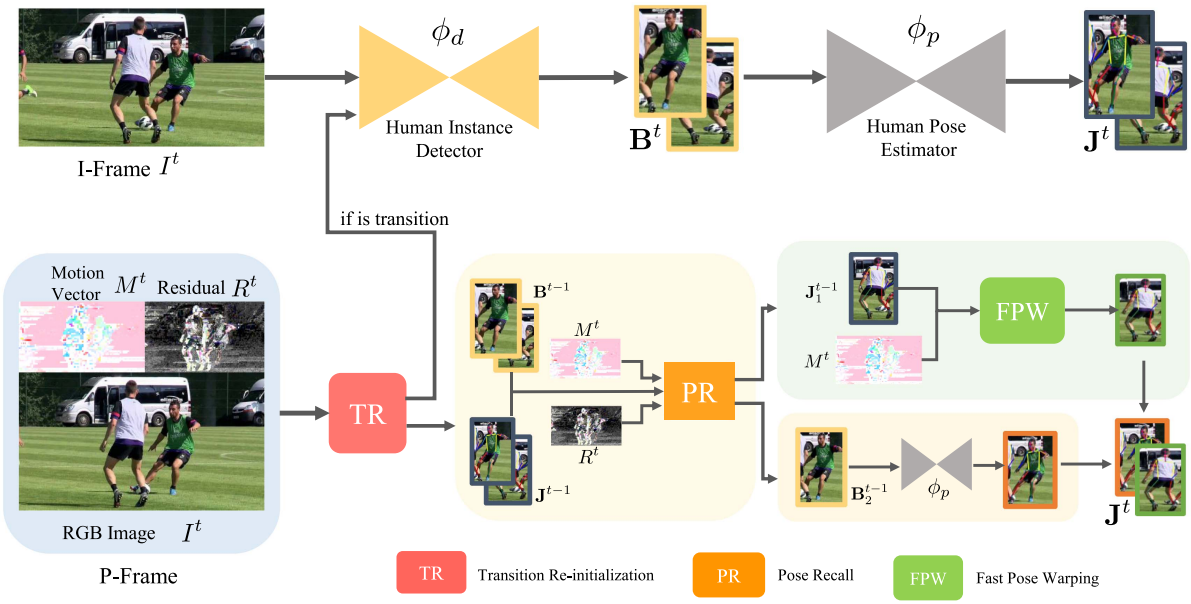

Fig. 2. An overview of our proposed method for fast human pose estimation in compressed videos. I-frames are directly sent to a human detector to detect each person. Then a human pose estimator is applied to each person instance to produce a corresponding human pose. For each P-frame, we first use the transition re-initialization module (TR). If a scene transition is detected, these frames would be treated the same as an I-frame by reconstructing its RGB image. Otherwise, each person instance in the P-frame is passed to the pose recall module (PR) to decide whether we need to re-initialize the pose estimation for this person. If a person instance passes the TR and PR modules, we can directly obtain its pose in the current P-frame by warping the pose joints from the previous frame using our fast pose warping (FPW) module.

the previous frame. By reusing the inference results from the previous frame, both modules can significantly speed up the pose estimation in P-frames. Although direct warping is fast, it is possible for the warping error to accumulate over time, and the tracking points gradually shift off the human body. In order to control the error propagation, we further design a pose recall module to correct the pose estimation results when the motion is too complex to follow. Another challenge to the fast warping approach is the occurrence of scene transition, which breaks the relationship between consecutive frames. To address this challenge, we design a transition re-initialization module to detect such scene transition so that the pose estimation can be re-initialized on the first frame of the new scene. Note that the PR and TR modules depend only on the compressed domain features and thus introduce minimal overhead into the whole pipeline.

Fig. 2 presents the complete data flow of the proposed framework when processing a compressed video. After decoding each GOP, the leading I-frame is first sent to the human instance detector and the pose estimator to obtain the location of body joints. Then the results of P-frames are efficiently predicted by the FWP module unless the PR and TR modules are triggered to re-initialize the pose estimation results of several human instances or the whole image.

The proposed framework exhibits three major advantages over the traditional per-frame framework. First, the proposed framework does not need to perform image-based pose estimation on most P-frames, resulting in a significant speedup on highly compressed videos. Second, all the additional modules rely only on the features that are readily available in a compressed video stream. So they introduce minimal overhead into the pipeline. Third, this framework is compatible with a wide range of image-based pose estimation methods and consistently achieves 2 to 5× speedup while achieving comparable accuracy.

We will discuss the details of each component below.

Human Instance Detector & Human Pose Estimator: We start by introducing the image-based pose estimation pipeline for the I-frames. Since an I frame is represented as a standard RGB image in a compressed video, we can choose any image-based human instance detector, denoted by ϕ_d and a pose estimator, denoted by ϕ_p, to initialize the human pose estimation in a GOP. In this paper, we adopt the HRNet [46], which uses an adapted Faster-RCNN for human detection and a specifically designed CNN for subsequent pose estimation. However, we emphasize that any pose estimation methods sharing a similar pipeline can be easily plugged into our proposed framework. We also conduct a study to illustrate the influence of different image-based pose estimators in Section V-C. In addition to operating on I-frames, ϕ_d and ϕ_p will also be used to re-initialize on a P-frame by reconstructing its RGB image if the TR or PR modules are triggered on this P-frame.

Fast Pose Warping: The FPW module is performed on each human instance to localize the joints of this person. Specifically, the module warps the human joints $\mathbf{J}_i^{t-1}$ of the i-th person in frame $\mathbf{I}^{t-1}$ with the motion vectors $\mathbf{M}^t$ at time t. It then generates a new set of joints location $\mathbf{J}_i^t$ for the same person by solving the following equations:

$$\mathbf{J}_i^t(n) - \mathbf{M}^t(\mathbf{J}_i^t(n)) = \mathbf{J}_i^{t-1}(n), n = 1, \ldots, N \qquad (2)$$

where $\mathbf{J}_i^t(n)$ denotes the coordinates of the n-th joint of the i-th person in frame $\mathbf{I}^t$, and where N indicates the total number of joints of this person.

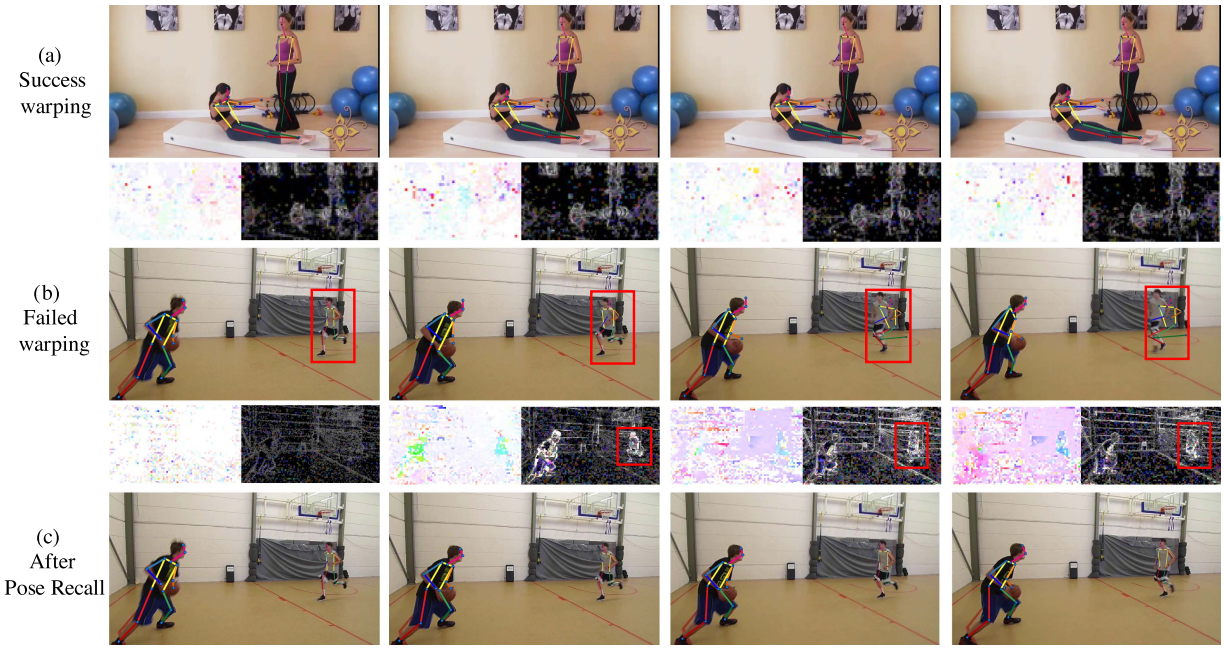

Fig. 3. a) and b) only adopt FPW to propagate pose from left to right. c) shows the results corrected by the PR module. Each column illustrates four consecutive frames in the same GOP.

Since the approximation of block-matching algorithm is usually adopted for calculating motion vector, motion vector often fails to associate the human parts of successive frames when there is severe motion (see Fig. 3(b)), which reflects in big magnitude on residual.

Pose Recall Module: To solve the problem of the loss of motion relation introduced by extensive and severe pose variance, we design the pose recall module. Before the pose recall, we firstly fast propagate the human bounding box with (2) by the center coordinates of the box. Then, for a given P-frame, the goal of this module is to decide whether the pose estimation results obtained from the pose warping are likely to be unreliable. If so, it will run the image-based pose estimator on a few specific human instances.

We design this module by considering the residual in each human instance and the motion information on human body joints to adaptively select the person with fast motion. Specifically, this module is based on two measures called the *motion intensity* and the *residual intensity* defined below.

The motion intensity is defined as the average motion on each body joint. It is computed as follows. For the i-th person in the current frame, we define the motion intensity (MI) of this person as the average motion magnitude on the joints:

$$MI_i = \frac{1}{2\,N}\sum_{n=1}^{N}(|\mathbf{M}_i^t(\mathbf{J}_i^t(n),0)| + |\mathbf{M}_i^t(\mathbf{J}_i^t(n),1)|) \tag{3}$$

Noted that $|.|$ is the absolute value operator and all the operations are element-wise. Here, 0 and 1 are the channels of the motion vector.

The residual map measures the error after warping the pixels in a P-frame using a motion vector. The absolute values in the residual map can be regarded as the confidence map of motion vectors. Larger values in the residual map tend to correspond to areas where the motion vectors are not reliable. We define the residual intensity as the average magnitude of the absolute value of residual (RI) for each human instance i.

$$RI_i = \frac{\sum_{(x,y)\in(H_i,W_i)}|\mathbf{R_i}(x,y)|}{H_i \times W_i} \tag{4}$$

where h_i and w_i denote the length and width of each human bounding box.

Then we select each person in the frame where the motion intensity or the residual intensity is above a certain threshold. Then the selected person instance is sent to the image-based pose estimator for re-initialization. Fig. 3 illustrates the benefit of adopting a pose recall module.

Transition Re-initialization: Some videos used in our dataset contain scene transitions due to camera switching. For the frame at the camera transition, the human pose in the current frame is often uncorrelated to the previous frame. As a result, the motion vector map does not provide any information for valid pose warping (see Fig. 4). This is especially problematic if the transition is at the beginning of a GOP. In this case, the unmatched human pose would be propagated to the remaining frames in this GOP. To address this issue, we propose the transition re-initialization module to specifically handle the camera transition.

Our key observation is that when the camera transition happens, the residual map of the corresponding frame tends to have enormous values. This is due to the fact that the two frames at the transition correspond to completely different scenes. Unlike the pose recall module that operates on each person in a frame, this module operates on the global information of the motion vector map. Once the residual intensity on the entire image is higher than a threshold THR_{trans}, we consider the frame to

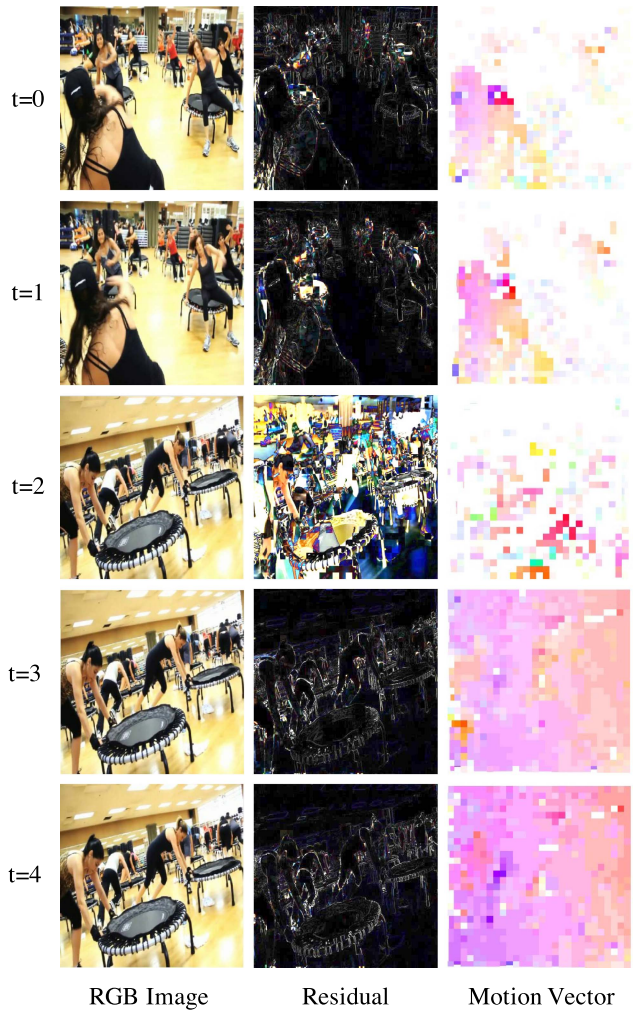

Fig. 4. Illustration of the motion vector and residual at transition. The residual can better indicate camera transition.

Algorithm 1: Overall Inference Algorithm.

Require: Pose estimator model ϕ_p, human detector model ϕ_d, compressed video stream $\mathbf{V}$ and fast pose warp operation ω.

Output : $\mathbf{J}^t$

for $t = 1$ *to* $|\boldsymbol{V}|$ **do**
  if *is I frame* **then**
    decode I frame to $\mathbf{I}^t$
    $B^t = \phi_d(I^t)$ # detect each person
    $\mathbf{J}^t = \phi_p(B^t)$ # estimate human pose
  else
    decode P frame to $\mathbf{M}^t, \mathbf{R}^t$ and $\mathbf{I}^t$
    if $RIT > THR_{trans}$ **then**
      $B^t = \phi_d(\mathbf{I}^t)$
      $\mathbf{J}^t = \phi_p(B^t)$
    else
      for $i = 1$ *to* $|\mathbf{J}^{t-1}|$ **do**
        if $MI_i^t \leq THR_{motion}$ *and* $RI_i^t \leq THR_{res}$ **then**
          $\mathbf{J}_i^t = \omega(\mathbf{J}_i^{t-1})$
        else
          $\mathbf{J}_i^t = \phi_p(B_i^t)$

TABLE I
THE DETAILS OF THE DATASETS USED IN THIS PAPER

Data Split	Train	Validation	Test	annotations/clip
PoseTrack17	250	50	214	30
PoseTrack18	593	170	375	30

We Illustrate the Number of Video Clips of Train, Val and Test Split. In Addition, annotations/clip Denotes the Number of Annotations Per Video Clip

be a transition. This frame is then sent to the pose estimator for re-initialization. The residual intensity of transition RIT is defined in a similar way as in (4), but the average is performed on the whole residual map instead of a human bounding box, i.e. $(H_i, W_i, \mathbf{R}_i)$ in (4) is replaced by $(H, W, \mathbf{R})$ for the frame. We show an example of the camera transition in Fig. 4.

The overall algorithm of our framework is shown in Algorithm 1. Note that our proposed three modules, *i.e.* FPW, PR, TR, are not built with neural networks. So these modules do not have additional model parameters.

V. EXPERIMENTS

In this section, we first describe the datasets and the implementation details. We then present ablation studies on various aspects of the proposed framework and compare it with other methods.

A. Dataset

PoseTrack [1] is a commonly used video-based benchmark for multi-person pose estimation and tracking. The videos from this dataset contain various challenging scenarios. For example, many videos include severe body motion, body pose variations, video transitions, highly occluded human instances and crowded scenes with dynamic human movements. These difficulties make it hard to achieve high accuracy on this dataset. PoseTrack has two different released datasets called PoseTrack17 and PoseTrack 18. PoseTrack17 contains in total 514 video sequences, in which 250, 50 and 214 clips are used as train, validation and test data, respectively. PoseTrack18 is significantly larger than PoseTrack17. The new release contains 593 train, 170 validation and 375 test clips, respectively. However, both of the two datasets only annotate 30 frames around the center of training clips. The annotations include head bounding boxes and 15 human key joins with indications on whether the joints are visible. The details of the two datasets are shown in Table I.

In this paper, we conduct ablation studies and experiments on both PoseTrack 17 and PoseTrack18 datasets using the official train, validation and test split. The human pose estimator is fine-tuned on the training set. We then evaluate our proposed framework on validation and test sets. The evaluation metric used in this work is the mean average precision (mAP) as in [1], [40].

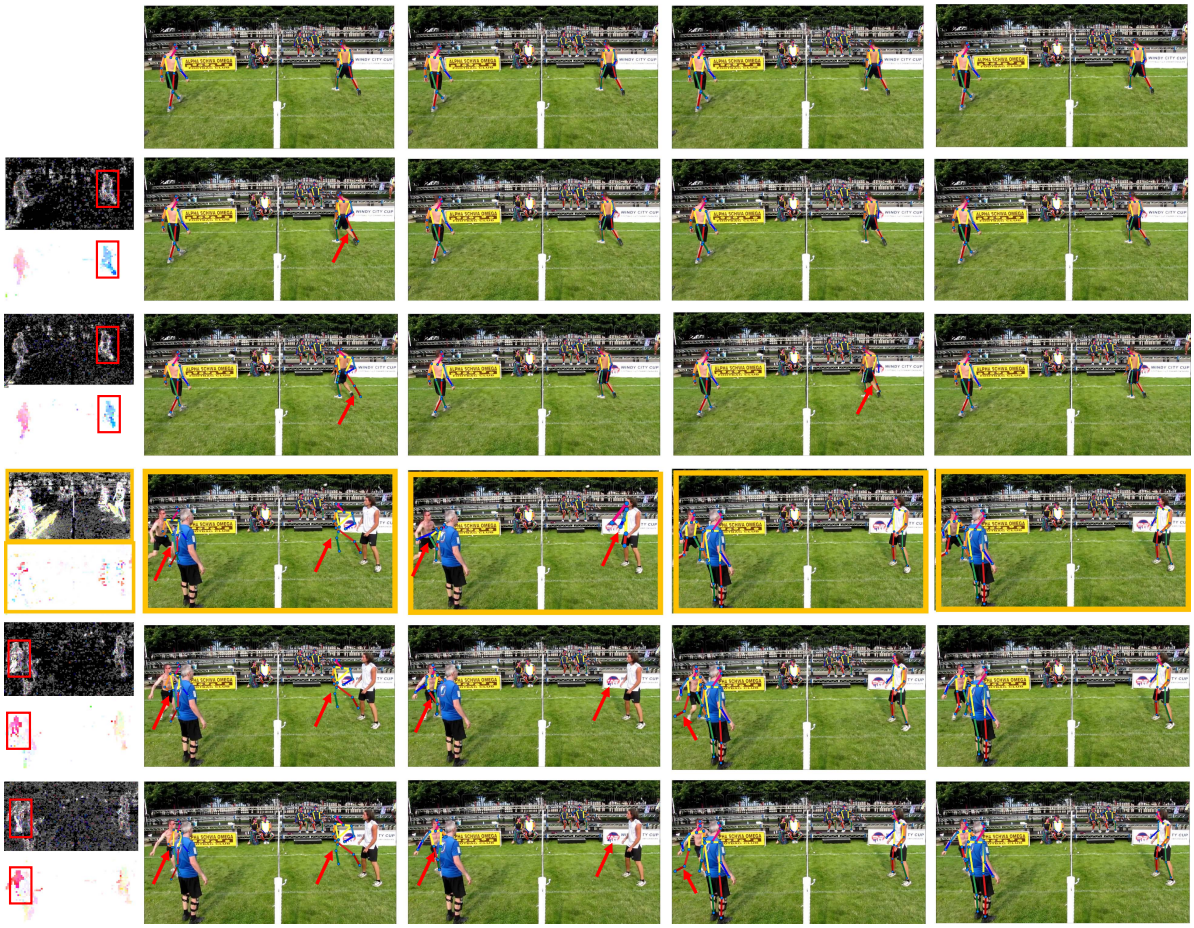

Fig. 5. Qualitative results of the ablation studies on the PoseTrack18 validation set. We use red arrows to point out the estimation error, orange boxes to indicate the camera transition and red boxes to illustrate the human instance being recalled by our PR module. From top to bottom, the decreased number of red arrows indicates the effectiveness of our modules.

B. Implementation Details

We choose HRNet-W48 [46] as our pose estimator. It is pre-trained on the COCO dataset and finetuned on the Posetrack18 training set. The finetuning process starts with an initial learning rate of 10^{-4} for 10 epochs. We then reduce the learning rate by a factor of 10 until the end of 20 epochs. For data augmentation, we take random samples uniformly distributed over $[-45°, 45°]$ and $[0.65, 1.35]$ for random rotation and random scale respectively. Flipping and half body data augmentation [49] are also used. We adopt the detector in [17] for human bounding box detection. The three thresholds, $THR_{trans}, THR_{motion}$ and THR_{res}, in Algorithm 1 are set to 3, 50, and 5 respectively. We use MPEG-2 Part2 (Simple Profile) [27] as our codec to compress the PoseTrack videos with the default GOP size 12. We do not use any data augmentation during testing. Our proposed method is implemented using Pytorch [36] and all the evaluations are conducted on the same Nvidia P100 GPU.

C. Ablation Studies

In this section, we perform extensive ablation studies on various aspects of the proposed framework. All the ablation studies are conducted on the PoseTrack 18 validation set.

TABLE II
ABLATION STUDIES ON THE EFFECTS OF EACH INDIVIDUAL MODULE ON BOTH ACCURACY AND SPEED

FPW	PR	TR	mAP	FPS
-	-	-	79.2	5.8
✓	-	-	54.1	49.2
✓	✓	-	72.6	22.5
✓	-	✓	71.8	36.1
-	✓	✓	77.3	5.4
✓	✓	✓	77.2	19.2

Effects of Individual Module: We perform ablation studies to demonstrate the effectiveness of each module in our proposed framework by removing one or more modules. The results are shown in Table II, from which we can make several observations: 1) the fast pose warping module can efficiently accelerate the human pose estimation with the off-the-shelf pose estimator; 2) the pose recall module can effectively identify significant

TABLE III
COMPARISON WITH POSE WARPING USING OPTICAL FLOW

Method	T_{flow}	T_{warp}	mAP
HRNet+FlowNet2	56 ms	7.6 ms	55.9
HRNet+FlowNet2s	18 ms	7.6 ms	53.9
HRNet+PWCNet	14 ms	7.6 ms	54.8
Ours(FPW only)	-	7.6 ms	54.1

We Experiment With Several Different Optical Flow Algorithms. T_{flow} Represents the Time for Estimating Optical Flow, and T_{warp} Denotes the Time for Warping. Our Method is More Efficient Since It Does Not Require Computing Optical Flow. At the Same Time, the Performance of Our Method is Comparable to Those Using Optical Flow for Pose Warping

TABLE IV
THE INFERENCE SPEED AND ESTIMATION ACCURACY WITH DIFFERENT GOP SIZES. THE FRAME SIZE IS SET TO 384*288

Method	GOP size	mAP	FPS
FPW only	1	79.2	5.8
Ours (HRNet)	1	79.2	5.8
FPW only	4	70.1	16.5
Ours (HRNet)	4	78.4	9.2
FPW only	8	61.2	33.1
Ours (HRNet)	8	77.8	14.7
FPW only	12	54.1	49.2
Ours (HRNet)	12	77.2	19.2

motion in videos and re-initialize the pose estimation of an individual person; 3) the transition re-initialization module can detect "hard-to-warp" frames and video transitions, which can avoid error propagation along the time sequence; 4) the entire framework with all these modules achieves the best overall balance between accuracy and speed.

Fig. 5 shows some qualitative examples of different methods. There is no surprise that the method with only the fast pose warping has the best efficiency. However, if we only use FPW, the accuracy degrades dramatically. Fig. 5(b) illustrates that the error is accumulated from the beginning of the GOP to the end. Especially on camera transition, the FPW module still propagates the unreliable pose to the next frame, resulting in inaccurate estimation. In general, human motion is exceptionally complicated. Directly warping poses with a motion vector could significantly jeopardize the performance. The method with pose recall and FPW solves the above problem to some extent. During inference, the person with a significant pose variance is rebooted and FPW terminates the error to be propagated to the next frame. From Fig. 5(c), we can observe that the PR module can avoid inaccurate pose warping before camera transition. However, after camera transition, the PR module cannot employ human pose estimation with inaccurate bounding boxes. Thus we can see the pose of the person with a white t-shirt is missing. We then show the performance of using both FPW module and TR module. It can be seen from Fig. 5(d) that transition re-initialization can greatly help boost the performance of FPW. Our method (Fig. 5(e)) using all modules gives the best qualitative results.

Warping by Optical Flow: We conduct a comparison with pose warping using optical flow instead of motion vectors. This will show the effectiveness of our proposed framework in terms of accelerating the inference speed. We have experimented with using PWCNet [45] and FlowNet2 [21] for optical flow estimation, respectively. Table III shows the performance of our method using FPW and optical flow-based methods. Surprisingly, we observe similar accuracy between using the motion vector and using optical flow. This phenomenon indicates that accurate motion modeling provided by optical flow does not provide additional benefit for propagating human pose across video frames compared with motion vectors that are already available in compressed videos. Another observation from Table III is that our fast feature warping is about 3-8 times faster than optical flow estimation. The gain on inference speed is mainly from the fact that we take the existing motion representations from the compressed video domain instead of relying on expensive optical flow estimation.

Effect of GOP Size: Table IV illustrates the performance of our method under different values of the GOP size. In order to demonstrate the significance of our method in balancing between inference speed and estimation accuracy, we show experimental results of the baseline method that only uses fast pose warping (FPW). When the GOP size is set to 1, the task of video pose estimation is degraded to per-frame pose estimation. In this case, the accuracy and inference speed of the two methods are the same. With the increase of the GOP size, we can generally see a decreasing trend in accuracy and an increasing trend in inference speed. However, benefiting from our PR and TR module, our method is less sensitive to the GOP size. The accuracy of our method only decreases from 79.2 to 77.2 with nearly 4 times speed-up. In contrast, only adopting the FPW module for fast warping causes the performance to drop significantly from 79.2 to 54.1. This ablation study further proves the robustness of our method.

Influences of Crop Size: The input image size also influences the inference speed. Intuitively, the inference speed can be accelerated as the input image becomes smaller. We conduct experiments to show the influence on our proposed method in terms of crop size for each human instance. We choose to crop human instances with two commonly used bounding box sizes (384×288 and 256×192). The performance of our method with two sizes is shown in Table VI. The inference speed increases when the input size of a human instance is decreased from 384×288 to 256×192. The gain on inference speed is mainly due to the fact that the human pose estimation model can run faster on a smaller input image. However, our overall framework can work with any input size. For example, our fast pose warping module only takes the joint coordinates from the previous frame and warps poses regardless of the size of each person.

TABLE V
QUANTITATIVE COMPARISON ON THE POSETRACK BENCHMARK

Dataset	Methods	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Mean	FPS
PoseTrack17 Val Set	PoseFlow [54]	66.7	73.3	68.3	61.1	67.5	67.0	61.3	66.5	10.2
	JointFlow [14]	-	-	-	-	-	-	-	69.3	-
	FastPose [56]	80.0	80.3	69.5	59.1	71.4	67.5	59.4	70.3	6.4
	SimpleBaseline [53]	81.7	83.4	80.0	72.4	75.3	74.8	67.1	76.7	6.2
	HRNet [46]	82.1	83.6	80.4	73.3	75.5	75.3	68.5	77.3	5.3
	PoseWarper [5]	81.4	88.3	83.9	78.0	82.4	80.5	73.6	81.2	1.9
	Ours	79.9	87.6	82.8	76.7	80.7	79.4	72.8	80.0	19.0
PoseTrack17 Test Set	PoseFlow [54]	64.9	67.5	65.0	59.0	62.5	62.8	57.9	63.0	9.7
	JointFlow [14]	-	-	-	53.1	-	-	50.4	63.4	-
	SimpleBaseline [53]	80.1	80.2	76.9	71.5	72.5	72.4	65.7	74.6	5.9
	HRNet [46]	80.1	80.2	76.9	72.0	73.4	72.5	67.0	74.9	5.8
	PoseWarper [5]	79.5	84.3	80.1	75.8	77.6	76.8	70.8	77.9	
	Ours	78.4	83.8	79.3	74.3	75.4	75.4	69.6	76.7	18.6
PoseTrack18 Val Set	AlphaPose [15]	63.9	78.7	77.4	71.0	73.7	73.0	69.7	71.9	14.8
	MDPN [18]	75.3	81.2	79.0	74.1	72.4	73.0	69.9	75.0	-
	PoseWarper [5]	79.9	86.3	82.4	77.5	79.8	78.8	73.2	79.7	2.3
	Ours	78.8	84.8	79.8	73.2	76.2	75.6	69.9	77.2	19.2
PoseTrack18 Test Set	AlphaPose++ [18], [15]	-	-	-	66.2	-	-	65.0	67.6	-
	MDPN [18]	-	-	-	74.5	-	-	69.0	76.4	-
	PoseWarper [5]	78.9	84.4	80.9	76.8	75.6	77.5	71.8	78.0	1.7
	Ours	76.8	82.4	78.2	73.0	71.5	74.6	69.0	75.2	18.8

The Performance of the Comparisions are Collected Either From PoseTrack Leaderboard or Paper. We Additionally Report the FPS of Methods That are Open-Sourced

TABLE VI
THE INFERENCE SPEED AND ESTIMATION ACCURACY WITH DIFFERENT SIZE OF INPUT IMAGES

Method	Input size	mAP	FPS
8-stage Hourglass	256*192	59.8	3.2
8-stage Hourglass	384*288	62.3	2.1
Ours (Hourglass)	256*192	58.1	13.6
Ours (Hourglass)	384*288	60.8	10.2
SimpleBaseline	256*192	75.6	11.2
SimpleBaseline	384*288	77.9	6.7
Ours (SimpleBaseline)	256*192	73.2	25.7
Ours (SimpleBaseline)	384*288	75.8	20.8
HRNet-W48	256*192	77.4	9.7
HRNet-W48	384*288	79.2	5.8
Ours (HRNet)	256*192	75.4	24.3
Ours (HRNet)	384*288	77.2	19.2

We Also Show Our Framework When Using Different Human Pose Estimators (HRNet, SimpleBaseline and 8-Stage Hourglass) for the Pose Estimation Module. Our Framework Can Always Significantly Accelerate Inference Speed Without Too Much Accuracy Drop

Effects of Image-based Pose Estimator: Our overall framework does not depend on the particular choice of the image-based pose estimator. In this experiment, we show the performance of our framework adopting three different state-of-the-art pose estimation models, *i.e*. simple baseline [53], HRNet [46] and 8-stage Hourglass [32]. The performance of using the three baselines is shown in Table VI. We can observe that these three pose estimation methods can be significantly accelerated once used within our framework. It is worth mentioning that we achieve 5 times speedup on 8-stage Hourglass. The empirical analysis further illustrates the advantage of our method for speeding up human pose estimation.

Overall Inference Speed: In this section, we investigate the overall inference speed of per-frame-based methods and our proposed one. We consider the timekeeping after the video is decompressed. Specifically, we add the inference time of the human detector in the pose estimation process. The comparison of overall inference speed is shown in Fig. 7. It can be noticed that our framework can be 3-5 times faster than the per-frame-based methods. One reason is that per-frame-based methods require human bounding box detection for every frame, while our framework only needs such detection on I-frames. In other words, with PR module, our method allows human detection on a subset of frames (*e.g*. I-frame) and quickly propagates bounding boxes from the current frame to other frames in a GOP. The results also provide a shred of solid evidence that our method is more efficient when deployed in practice.

D. Main Results

Table V shows the quantitative comparison of our approach with several state-of-the-art human pose estimation methods in terms of accuracy and speed. The comparison is conducted on both PoseTrack17 and PoseTrack18 datasets. We can generally

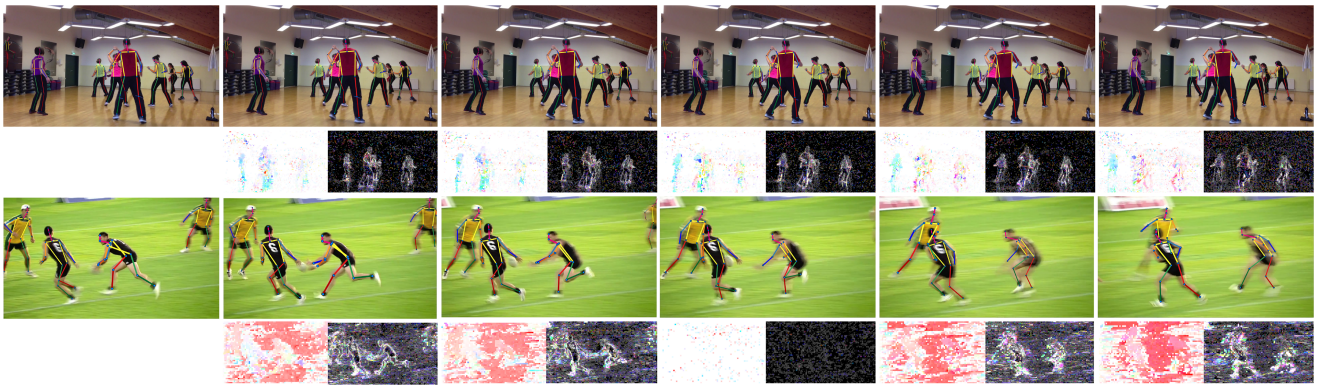

Fig. 6. Qualitative results on the PoseTrack18 validation set. The first column corresponds to I-frames, while other columns correspond to P-frames in a GOP.

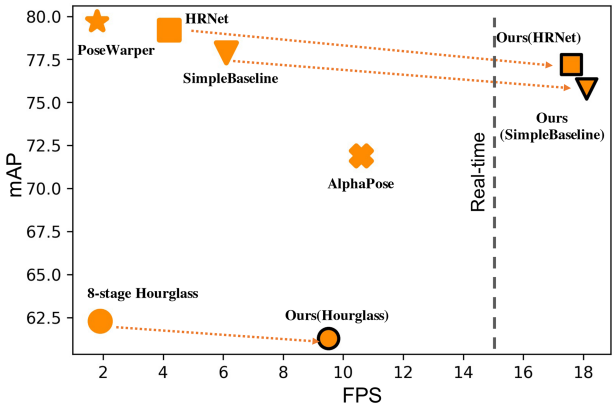

Fig. 7. Illustration of overall running time on the PoseTrack18 validation dataset. The overall running time consists of bounding box proposal time and human pose estimation time.

Fig. 8. Failure cases are pointed by red arrows. The pose of elbow and head are unable to be detected because of occlusion and image blurry.

observe a trade-off between inference speed and accuracy in Table V. For example, although PoseWarper is the top-performing method for all three datasets, the inference speed is the slowest. PoseFlow and AlphaPose can run over ten frames per second. However, the accuracy of the two methods is 10 mAP less than the top performed methods. Somewhat surprisingly, our method is the only one that can estimate human pose in real-time over 19 FPS, while achieving accuracy comparable to the top-performing method. It is worth mentioning that our performance is better than the original HRNet. We show some qualitative examples of our method in Fig. 6.

E. Limitation and Future Works

Due to the fact that our proposed approach is introduced to accelerate the current per-frame human pose estimation method, we notice that our proposed method might inherit the limitation of per-frame-based methods. Fig. 8 shows some failure cases of our method. It can be observed that our method fails to predict accurate human pose when the human joint is occluded or the image is blurry. The two problems are also the main challenges in per-frame-based human pose estimation [59]. It is preferable to address the problems in the future.

VI. Conclusion

In this paper, we have introduced the task of human pose estimation in the compressed video domain. The goal is to take advantage of the motion representation (*i.e.* motion vectors) that is already encoded in a video stream to accelerate the pose estimation. The proposed framework uses motion vectors to propagate the estimated pose joints from the I-frame to other P-frames. We also introduce additional modules to re-initialize the pose estimation when the pose propagation is unreliable due to large motions or scene transition. Overall, our proposed framework achieves a nice balance between accuracy and inference speed.

References

[1] M. Andriluka *et al.*, "Posetrack: A benchmark for human pose estimation and tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5167–5176.
[2] M. Andriluka, S. Roth, and B. Schiele, "Pictorial structures revisited: People detection and articulated pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 1014–1021.
[3] M. Andriluka, S. Roth, and B. Schiele, "Monocular 3D pose estimation and tracking by detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 623–630.
[4] A. Barjatya, "Block matching algorithms for motion estimation," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 225–239, Apr. 2004.
[5] G. Bertasius, C. Feichtenhofer, D. Tran, J. Shi, and L. Torresani, "Learning temporal pose estimation from sparsely-labeled videos," *Adv. Neural Inf. Process. Syst.*, 2019.
[6] A. Bulat and G. Tzimiropoulos, "Binarized convolutional landmark localizers for human pose estimation and face alignment with limited resources," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 3706–3714.

[7] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2D pose estimation using part affinity fields," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1302–1310,.

[8] A. Chadha, A. Abbas, and Y. Andreopoulos, "Compressed-domain video classification with deep neural networks:"There's way too much information to decode the matrix"," in *Proc. IEEE Int. Conf. Image Process.*, 2017, pp. 1832–1836.

[9] A. Chadha, A. Abbas, and Y. Andreopoulos, "Video classification with CNNs: Using the codec as a spatio-temporal activity sensor," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 2, pp. 475–485, Feb. 2019.

[10] X. Chen and A. L. Yuille, "Articulated pose estimation by a graphical model with image dependent pairwise relations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1736–1744.

[11] Y. Chen *et al.*, "Cascaded pyramid network for multi-person pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7103–7112.

[12] Y.-M. Chen, I. V. Bajić, and P. Saeedi, "Moving region segmentation from compressed video using global motion estimation and Markov random fields," *IEEE Trans. Multimedia*, vol. 13, pp. 421–431, 2011.

[13] B. Cheng *et al.*, "HigherHRNet: Scale-aware representation learning for bottom-up human pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 5386–5395.

[14] A. Doering, U. Iqbal, and J. Gall, "Joint flow: Temporal flow fields for multi person tracking," in *Proc. Brit. Mach. Vis. Conf.*, 2018, p. 261.

[15] H.-S. Fang, S.Xie, Y.-W. Tai, and C. Lu, "RMPE: Regional multi-person pose estimation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2353–2362.

[16] J. Feng *et al.*, "TapLab: A fast framework for semantic video segmentation tapping into compressed-domain knowledge," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published, doi: 10.1109/TPAMI.2020.3024646.

[17] R. Girdhar, G. Gkioxari, L. Torresani, M. Paluri, and D. Tran, "Detect-and-track: Efficient pose estimation in videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 350–359.

[18] H. Guo *et al.*, "Multi-domain pose network for multi-person pose estimation and tracking," in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2018.

[19] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2961–2969.

[20] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.

[21] E. Ilg *et al.*, "FlowNet 2.0: Evolution of optical flow estimation with deep networks," in *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, 2017, pp. 1647–1655.

[22] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele, "Deepercut: A deeper, stronger, and faster multi-person pose estimation model," in *Proc. Eur. Conf. Comput. Vis.*, Cham: Springer, 2016, pp. 34–50.

[23] S. Johnson and M. Everingham, "Clustered pose and nonlinear appearance models for human pose estimation," in *Proc. Brit. Mach. Vis. Conf.*, vol. 2, 2010, p. 5.

[24] A. Kamel, B. Sheng, P. Li, J. Kim, and D. D. Feng, "Hybrid refinement-correction heatmaps for human pose estimation," *IEEE Trans. Multimedia*, vol. 23, pp. 1330–1342, 2020.

[25] S.-T. Kim and H. J. Lee, "Lightweight stacked hourglass network for human pose estimation," *Appl. Sci.*, vol. 10, no. 18, pp. 6497, 2020.

[26] A. Krull *et al.*, "Learning analysis-by-synthesis for 6D pose estimation in RGB-D images," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 954–962.

[27] D. L. Gall, "MPEG: A video compression standard for multimedia applications," *Commun. ACM*, vol. 34, no. 4, pp. 46–58, 1991.

[28] S.-W. Lee, Y.-M.Kim, and S. W. Choi, "Fast scene change detection using direct feature extraction from MPEG compressed videos," *IEEE Trans. Multimedia*, vol. 2, pp. 240–254, 2000.

[29] A. Li, Y. Lu, and Y. Wang, "Semantic segmentation in compressed videos," in *Proc. IEEE 21st Int. Workshop Multimedia Signal Process.*, 2019, pp. 1–5.

[30] M. Li, Z. Zhou, and X. Liu, "Multi-person pose estimation using bounding box constraint and LSTM," *IEEE Trans. Multimedia*, vol. 21, pp. 2653–2663, 2019.

[31] D. Mukherjee *et al.*, "A technical overview of VP9-the latest open-source video codec," *SMPTE Mot. Imag. J.*, vol. 124, no. 1, pp. 44–54, 2015.

[32] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Proc. Eur. Conf. Comput. Vis.*, Cham: Springer, 2016, pp. 483–499.

[33] G. Ning, Z. Zhang, and Z. He, "Knowledge-guided deep fractal neural networks for human pose estimation," *IEEE Trans. Multimedia*, vol. 20, pp. 1246–1259, 2017.

[34] W. Ouyang, X. Chu, and X. Wang, "Multi-source deep learning for human pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 2337–2344.

[35] G. Papandreou *et al.*, "Towards accurate multi-person pose estimation in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3711–3719.

[36] A. Paszke *et al.*, "Automatic differentiation in pytorch," *Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.

[37] S.-C. Pei and Y.-Z. Chou, "Effective wipe detection in MPEG compressed video using macro block type information," *IEEE Trans. Multimedia*, vol. 4, pp. 309–319, 2002.

[38] T. Pfister, J. Charles, and A. Zisserman, "Flowing convnets for human pose estimation in videos," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1913–1921.

[39] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele, "Strong appearance and expressive spatial models for human pose estimation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 3487–3494.

[40] L. Pishchulin *et al.*, "DeepCut: Joint subset partition and labeling for multi person pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4929–4937.

[41] U. Rafi, B. Leibe, J. Gall, and I. Kostrikov, "An efficient convolutional network for human pose estimation," in *Proc. Brit. Mach. Vis. Conf.*, vol. 1, 2016, p. 2.

[42] M. Schwarz, H. Schulz, and S. Behnke, "RGB-D object recognition and pose estimation based on pre-trained convolutional neural network features," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 1329–1335.

[43] J. Song, L. Wang, L. V. Gool, and O. Hilliges, "Thin-slicing network: A deep structured model for pose estimation in videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5563–5572.

[44] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[45] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8934–8943.

[46] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5686–5696.

[47] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler, "Joint training of a convolutional network and a graphical model for human pose estimation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1799–1807.

[48] A. Toshev and C. Szegedy, "DeepPose: Human pose estimation via deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1653–1660.

[49] Z. Wang *et al.*, "Mscoco keypoints challenge 2018," in *Proc. Joint Recognit. Challenge Workshop ECCV*, vol. 5, 2018.

[50] S.-E.Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4724–4732.

[51] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H. 264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[52] C.-Y.Wu *et al.*, "Compressed video action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6026–6035.

[53] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 466–481.

[54] Y. Xiu, J. Li, H. Wang, Y. Fang, and C. Lu, "Pose flow: Efficient online pose tracking," *Brit. Mach. Vis. Conf.*, 2018, p. 53.

[55] D. Zhang, G. Guo, D. Huang, and J. Han, "PoseFlow: A deep motion representation for understanding human behaviors in videos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6762–6770.

[56] F. Zhang, X. Zhu, and M. Ye, "Fast human pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3512–3521.

[57] G. Zhang, J. Liu, H. Li, Y. Q. Chen, and L. S. Davis, "Joint human detection and head pose estimation via multistream networks for RGB-D videos," *IEEE Signal Process. Lett.*, vol. 24, no. 11, pp. 1666–1670, Nov. 2017.

[58] Z. Zhang, J. Tang, and G. Wu, "Simple and lightweight human pose estimation," 2019, *arXiv:1911.10346*.

[59] C. Zheng *et al.*, "Deep learning-based human pose estimation: A survey," 2020, *arXiv:2012.13392*.