Tracking People with 3D Representations

Anonymous Author(s)
Affiliation
Address
email

Abstract

We present a novel approach for tracking multiple people in video. Unlike past approaches which employ 2D representations, we focus on using 3D representations of people, located in three-dimensional space. To this end, we develop a method, Human Mesh and Appearance Recovery (HMAR) which in addition to extracting the 3D geometry of the person as a SMPL mesh, also extracts appearance as a texture map on the triangles of the mesh. This serve as a 3D representation for appearance that is invariant to viewpoint and pose. Given a video clip, we first detect bounding boxes corresponding to people, and for each bounding box, we extract 3D appearance, pose, and location information using HMAR. These embedding vectors are then sent to a transformer, which performs spatio-temporal aggregation of these representations over the duration of the sequence. The similarity of the resulting representations is used to solve for associations that assigns each person to a tracklet. We evaluate our approach on the Posetrack, MuPoTs and AVA datasets. We find that 3D representations are more effective than 2D representations for tracking in these settings, and we obtain state-of-the-art performance.

1 Introduction

Humans are three-dimensional beings, who live and move in a three-dimensional space. However, people tracking algorithms from the computer vision community, e.g. [4, 6, 11, 27, 34, 39, 43] are typically based on 2D representations of people in 2D space. This made a lot of sense in the 1990s when people tracking first came to the fore as (1) the application context was often video surveillance, where people are at low resolutions (2) there weren't reliable algorithms for lifting people from 2D to 3D. But today neither of these limitations hold. Applications of video analysis can use HD video (1080p or better), and we have excellent technology e.g. [16] for 3Dfying people.

Figure 1 illustrates why tracking people might be easier in 3D than in 2D. Even when two people overlap and occlude each other in 2D, their spatial location is typically separable in 3D. While 2D appearance features are sensitive to viewpoint and body pose, 3D appearance features are invariant to both viewpoint and pose. These properties enable robust tracking through people-to-people occlusion and even shot changes, a common phenomena in edited media and online videos.

Tracking people in 3D also opens up many interesting downstream tasks such as predicting 3D human motion from video [17, 20], predicting their future behavior [10, 45], and imitating human behavior from video [29]. These tasks involve videos of people other than pedestrians, such as Internet videos of people performing sports, dance, daily activities [9], or even edited media such as movies or TV shows [12, 28].

Our input is a monocular video with detected bounding boxes. On each bounding box, we develop an approach that extracts both 3D geometry and appearance information of the person. Specifically, we extend HMR [16], a method that predicts 3D human mesh from monocular images, to also extract the appearance information of the person by predicting a flow field that estimates the texture map, which

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

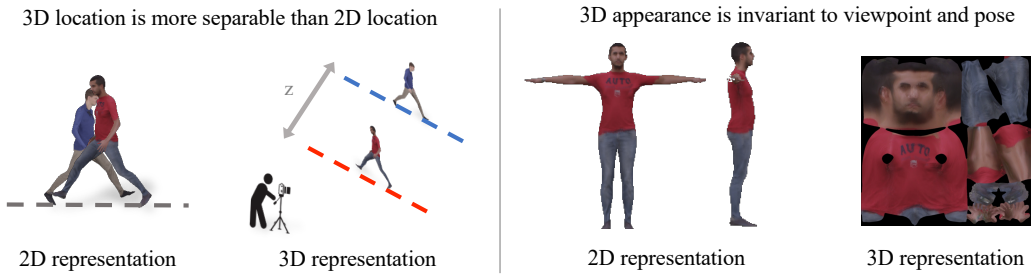

Figure 1: *The benefits of tracking people in 3D.* Left: Using the 3D location of people makes it easier to separate them than using their 2D location or 2D bounding boxes. Right: The appearance of people, when computed in a 3D representation (texture map) is invariant to viewpoint and pose.

is a viewpoint and pose invariant space. We refer to this model as Human Mesh and Appearance Recovery (HMAR). From HMAR output, we extract 3D appearance, 3D pose, joints, and estimate of their 3D location as separate features. We then aggregate this per-bounding box 3D information across time and space via a transformer [36] that takes these separate 3D cues as input. The transformer acts as a spatio-temporal diffusion mechanism that can propagate information across similar features by means of attention. Finally, we associate the diffused features across time to assign identities to each bounding box, which produces the final tracklets. The overview of our approach is shown in Figure 2.

We evaluate our algorithm on three different datasets PoseTrack [1], MuPots [26] and AVA [12]. AVA is a particularly interesting dataset as it is derived from movies, which include shot changes. These exhibit much greater variety of behavior than videos in the traditional tracking challenges such as MOT. In addition, people are larger (in terms of number of image pixels), which is essential for the 3D approach to work. We find the following: 1. 3D representations are more effective than 2D representations, 2. Ablation studies show that using 3D information in both appearance and location is helpful, 3. We outperform current state-of-the-art on these datasets.

2 Related work

Tracking The tracking literature is vast and we refer readers to [6] for a comprehensive summary. Tracking may be designed for generic categories such as any bounding boxes [49], here we focus on methods that track humans. These approaches mostly depend on 2D location or keypoint features [11, 34, 39] and sometimes 2D appearance [4, 27, 41, 43]. Most modern-day tracking approaches focus on evaluating their performance on the MOT benchmark [6], which mainly focus on high-density pedestrians scenes observed in outdoor or indoor settings. Under this setting, detection plays a key role and many methods jointly solve for both detection and association [4, 27]. In this work, we are interested in the effectiveness of 3D representation for tracking and thus assume that detection bounding boxes are provided, which we associate through our representations. Furthermore, we are interested in tracking people in every day Internet videos where people occupy larger portion of the scene, performing activities such as sports or dancing, and even edited media such as movies.

There are methods that incorporate 3D information in tracking, however these approaches assume multiple input cameras [22, 46] or 3D point cloud observation from lidar data [38]. Here we focus on the setting where the input is a monocular video.

Monocular 3D Human Reconstruction There is long history of methods that predict 3D human structure or motion from monocular images or videos [5, 13]. Recently, there has been rapid progress in this area due to the emergence of a statistical model of human bodies such as SMPL [24] that provides a low dimensional parameterization of a deformable 3D mesh of human bodies. Human Mesh Recovery (HMR) method [16] and follow-up works, e.g., [3, 21], employ convolutional neural networks to predict the parameters of an existing body model from a single image. Extending this result in the temporal dimension, HMMR [18] and VIBE [20] reconstruct mesh output for a single person given a series of tracked frames or tracklets as the input. Other works focus on multiple people [15, 44, 35] and regress the parameters corresponding to multiple people on the same input frame. All of these methods rely on detected bounding boxes or tracklets in the case of temporal domain. As such, these approaches rely heavily on existing tracking algorithms to extract long

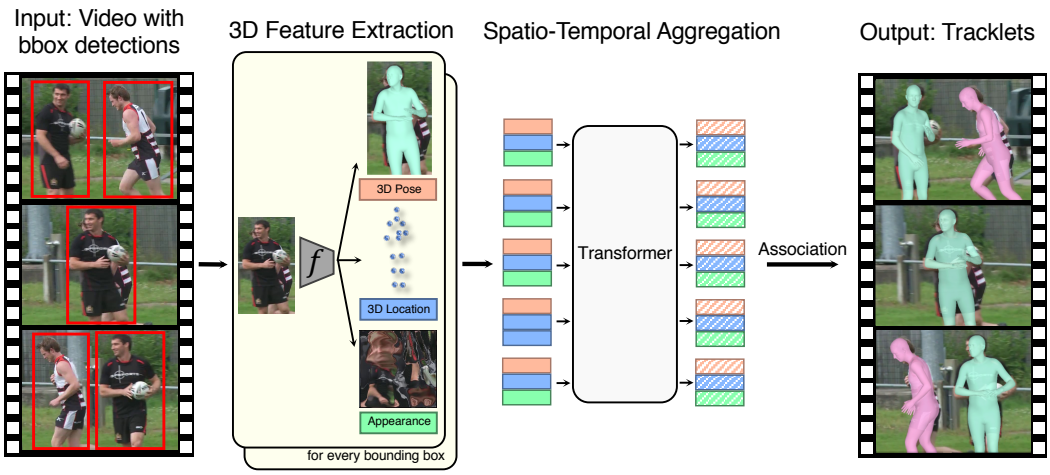

Figure 2: *Overview of our framework.* The input to our pipeline consists of a video sequence with bounding box detections. For each bounding box, using HMAR model we extract 3D representations corresponding to the person's 3D appearance, pose, and location. The 3D representations for all bounding boxes are then processed jointly by a transformer which aggregates information across space and time. Finally, an online association step assigns identity to each detected human, which constitute the tracklets.

tracklets associated to the same individual. In this work we explore how the predicted 3D human information may be used to empower tracking in exchange. There are two recent works that jointly solve tracking and 3D pose estimation of multiple people from monocular video [25, 31]. While these approaches focus on the task of 3D pose estimation, we focus on the problem of identity association and study the effectiveness of 3D vs 2D representation and compare extensively against state-of-the-art 2D tracking methods.

Related to our approach are recent methods that extract the texture of a person along with their geometry [37, 42]. These approaches use identity labels to learn an encoder that predicts complete 3D appearance expressed as UV images. In this work we explore how UV images can be used for associating people across time for tracking.

3 Method

In order to assess the effectiveness of 3D information in tracking, we propose a simple tracking framework that assumes as input a monocular video and a set of detected bounding boxes. We then explore how to associate these bounding boxes across time using various cues.

Our tracking algorithm consists of two main modules: our proposed HMAR model, which encodes humans into a rich embedding space, and a Transformer model for learning associations between detected humans across multiple frames. The HMAR model learns to predict appearance, pose, and keypoints. These are essential cues for tracking, and each of them carries complementary information. For example, when a person is temporarily occluded, the appearance is important to establish its identity after re-appearance, while when many people share similar clothing in a video, pose and location become the primary cues for tracking. Once, we have a rich embedding of a person, we need to learn associations between different human identities so that each person can be matched in the upcoming frames. We train a modified version of a transformer [36] to learn these associations across multiple human identities.

3.1 Human Mesh and Appearance Recovery (HMAR)

Our starting point for 3D feature extraction is the popular HMR architecture [16]. Given a bounding box of a person, HMR regresses the 3D pose and shape parameters of the person. While the 3D pose provides rich geometric information about the person, it throws away the appearance information, which is essential for tracking. To augment this, we extend HMR such that it can also recover the 3D

appearance of the person by means of a texture image, which is a space that is viewpoint and pose invariant. We call this method HMAR, which stands for Human Mesh and Appearance Recovery.

In order to recover the texture image, we create skip connections from the spatial features of the Resnet-50 backbone used in HMR and connect them to a new appearance head that maintains the spatial alignment of features through a series of convolution and bilinear upsampling layers (Figure 3). Please see the supplemental for details of the architecture. The appearance head takes these spatial features over multiple scales as input and is trained to predict the texture of the person, using the canonical SMPL texture mapping [24]. This is done in the form of appearance or texture flow in the spirit of [17, 48]. Specifically, we regress a texture flow $\mathcal{F}$ that has the same dimensions as the texture map. Each value of this flow indicates the coordinates (x, y) of the input image that the corresponding texel is sampled from. Eventually, the final texture map $I_{\text{tex}} = G(I, \mathcal{F})$ is produced by bilinear sampling G of the input image I. The predicted texture map is then used to render the textured 3D mesh. The model is trained to minimize the reconstruction loss after rendering the texture and geometry on the foreground pixels using L1 and perceptual loss.

3.2 3D Representations for Tracking

The tracking performance is greatly affected by the cues and the type of representation used for association. Our embeddings are computed as natural outputs of the HMAR model as follows:

Appearance: HMAR recovers the appearance of the person in the form of the texture image I_{tex}. For tracking, we need to create a compact embedding for appearance, so we employ an auto-encoder to further encode this appearance. The auto-encoder takes the texture image as input and is trained to reconstruct it by going through a low dimensional bottleneck. This bottleneck is represented by a 512-dimensional vector $\bar{\mathbf{a}}$, which is used as the representation of the appearance in our pipeline.

3D Pose: For the pose representation, we follow previous work [18, 45] and use the embedding $\bar{\mathbf{p}} \in \mathbb{R}^{2048}$ of HMR as a representation of the pose of the person.

3D Location: HMAR is given an image with dimensions $[W, H]$ and is applied on a square bounding box with center $[c_x, c_y]$ and dimensions $[b, b]$. HMAR also predicts the local camera parameters $[s, t_x, t_y]$ (scale and 2D translation) for the input bounding box. By assuming a constant focal length f for the image, we can recover the approximate 3D global translation of the person, which localizes the mesh in the view coordinate 3D space corresponding to the full image. This translation is expressed as:

$$T = \left[\; t_x + \frac{2c_x - W}{sb}, \quad t_y + \frac{2c_y - H}{sb}, \quad \frac{2f}{sb} \; \right]. \tag{1}$$

Since we do not know the actual focal length, this is only an approximation of their metric location, but this approach can recover a consistent relative 3D positions of the people over a single video up to scale.

For a more detailed description of the 3D location, we also extract the 3D keypoints of the person that are directly regressed from the recovered mesh, and we position them in the view coordinate 3D space by adding the translation T to them. This representation is strictly more informative than the 2D bounding box information, both because it is in 3D, and because it provides a localized detailed information about the pose of the person than a single bounding box. We also concatenate the temporal information to the 3D keypoints in the spirit of [36], which leads to space-time representation $\bar{\mathbf{s}} \in \mathbb{R}^{90}$.

3.3 Spatio-temporal Feature Aggregation

Our HMAR model encodes the 3D appearance, pose, and location information of each bounding box. While this gives us a rich 3D embedding, this representation only captures information from a single-frame. In order to incorporate the spatio-temporal information of the surrounding bounding boxes, we employ a modified Transformer model to aggregate global information across space and time. This transformer essentially performs anisotropic diffusion [30] of the representation across all bounding boxes, where the similarity is captured by attention.

Specifically, the input to transformer is the combination of the three different representations obtained from HMAR corresponding to 3D appearance, pose and location, $\mathbf{h} = [\bar{\mathbf{a}}^T, \bar{\mathbf{p}}^T, \bar{\mathbf{s}}^T]^T \in \mathbb{R}^{2650 \times 1}$. The transformer architecture consists of 3 blocks of self-attention and feed-forward layers. We use a

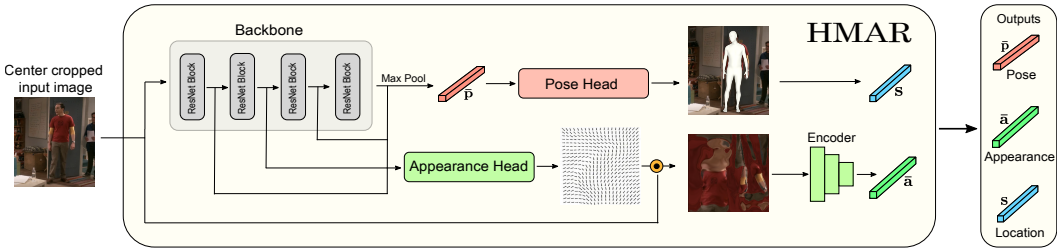

Figure 3: *Human Mesh and Appearance Recovery (HMAR):* Our HMAR model learns to predict 3D pose, a UV-map and a set of keypoints relative to the whole image from a single person center cropped image. For the pose, we use the HMR model [16], which gives us a 2048-dimensional pose vector ($\bar{\mathbf{p}}$). We add an additional appearance head to HMR, which learns to predict a flow-field. When the input image is sampled using the flow-field, it produces a UV-map for the appearance of the person in the image. The UV-map is then encoded into a appearance vector ($\bar{\mathbf{a}}$). Finally, we extract the location of 15 3D keypoints ($\mathbf{s}$) from the 3D mesh. These keypoints represent the position of the human in the 3D space. Therefore, using our HMAR model we can get pose, appearance, and location information of the person.

Figure 4: *HMAR results.* For each example we show from left to right: (a) Input image, (b) Predicted mesh, (c) predicted texture map, (d) and (e) textured mesh from frontal and side view.

single head attention inside the self-attention layer. During training, we feed T number of frames, each with a maximum number of P people on it. Therefore, for a single clip, the transformer could see $T \times P$ number of humans.

We use a modified transformer that treats each of the three attributes of the $\mathbf{h}^{t,i}$ separately. Specifically, inside the self-attention layer, appearance, pose and keypoints have their own value, query, and key vectors.

$$\mathbf{v}^{t,i}_{app} = W^v_{app}\bar{\mathbf{a}}, \qquad \mathbf{v}^{t,i}_{pose} = W^v_{pose}\bar{\mathbf{p}}, \qquad \mathbf{v}^{t,i}_{loc} = W^v_{loc}\bar{\mathbf{s}} \tag{2}$$

Here, $W^v_{app} \in \mathbb{R}^{512\times512}, W^v_{pose} \in \mathbb{R}^{2048\times2048}$ and $W^v_{loc} \in \mathbb{R}^{90\times90}$ are the transformation matrices for the value vectors of each attributes in $\mathbf{h}^{t,i}$ vector. Similarly, we can get $[\mathbf{q}^{t,i}_{app}, \mathbf{q}^{t,i}_{pose}, \mathbf{q}^{t,i}_{loc}]$ and $[\mathbf{k}^{t,i}_{app}, \mathbf{k}^{t,i}_{pose}, \mathbf{k}^{t,i}_{loc}]$ vectors by learning projections W^q_{att}, W^k_{att}, for $att \in [pose, loc]$. With this setting, we can find attentions for each attributes separately. Finally the total attention $\mathtt{A} \in \mathbb{R}^{TP\times TP}$ is taken as a weighted sum over various attentions. See fig 5.

$$\mathtt{A}^{(t,i),(t',i')} = \sum_{att\in[app,pose,loc]} \beta_{att} \cdot \mathtt{softmax}\left(\frac{\mathbf{q}^{t,i}_{att}(\mathbf{k}^{t',i'}_{att})^T}{\sqrt{dim_{att}}}\right) \tag{3}$$

Here, $\beta_{app}, \beta_{pose}$ and β_{loc} are a set of hyper-parameters. While we use equal values for all of our experiments for simplicity, this design allows for selective attention to different attributes. For example, if we use $[0, 1, 0]$ for $\beta_{app}, \beta_{pose}$ and β_{loc} respectively, the appearance and location embeddings are accumulated based only on the attention of the pose embedding. This could be useful for extreme cases, such as when where all people wear the same uniform. Finally, the output of the self-attention layer $\mathbf{h}^{t,i}_{sa}$ is calculated as the weighted sum of the total attention.

$$\mathbf{h}^{t,i}_{sa} = \mathbf{h}^{t,i} + \sum_{t',i'} \mathtt{A}^{(t,i),(t',i')} \cdot [\mathbf{v}^{t',i'}_{app}, \mathbf{v}^{t',i'}_{pose}, \mathbf{v}^{t',i'}_{loc}]^T \tag{4}$$

We use the same strategy for following feed-forward layers and normalization layers, by treating each attribute separately as in Eq 2. After these human vectors $\mathbf{h}^{t,i}$ have been processed by the transformer, the output vector $\widehat{\mathbf{h}}^{t,i}$ will accumulate information across multiple people over space-time. However, ideally we want $\widehat{\mathbf{h}}^{t,i}$ to attend only the persons with same identities. For example, $\widehat{\mathbf{h}}^{t,i}$ should attend $\widehat{\mathbf{h}}^{t',i'}$ if and only if i, i' are the same person in different frames. If this happens, $\widehat{\mathbf{h}}^{t,i}$ and $\widehat{\mathbf{h}}^{t',i;}$ will

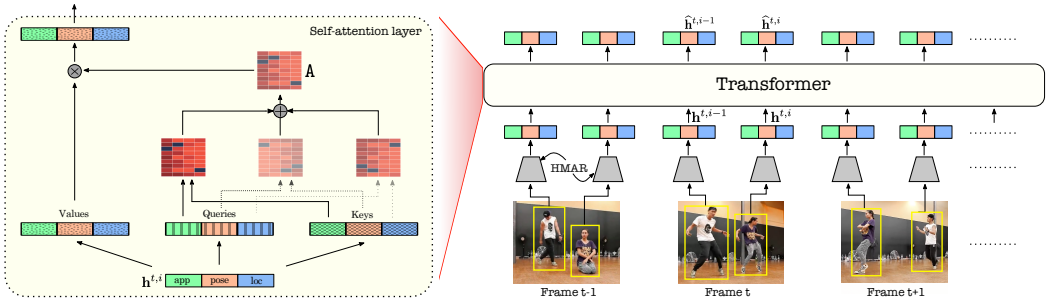

Figure 5: *Spatio-Temporal Aggregation with Transformer:* For T frames in a video, we obtain a set of bounding boxes from an off-the-shelf human detector [32]. We use HMAR to extract representations of 3D appearance, pose, and location from each bounding box, whose concatenation serve as a token for each bounding box. Since these information is only extracted from a single image, we send the tokens through a transformer, which acts as a diffusion mechanism to propagate information between similar tokens across space and time. We employ a modified transformer that computes attention on each embedding separately, which then gets merged together for a final attention map.

be the same or at least close to each other in some similarity metric $d(\widehat{\mathbf{h}}^{t,i}, \widehat{\mathbf{h}}^{t',i;})$. We enforce this during training by minimizing the ReID (ReIdentification) loss.

$$L_{\text{ReID}} = \sum_{t,t' \sim [1,2,\ldots,T]} d(\widehat{\mathbf{h}}^{t,i}, \widehat{\mathbf{h}}^{t',i'}) + \max\{0, m - d(\widehat{\mathbf{h}}^{t,i}, \widehat{\mathbf{h}}^{t',j'})\} \tag{5}$$

Here, $\widehat{\mathbf{h}}^{t,i}$ and $\widehat{\mathbf{h}}^{t',i'}$ are the embedding vectors for the same person in frame t, t', while $\widehat{\mathbf{h}}^{t,i}$ and $\widehat{\mathbf{h}}^{t',j'}$ are the embedding vectors for two different people in frames t, t'. $\widehat{\mathbf{h}}^{t,i}$ act as a anchor for the contrastive loss. We use margin $m = 10$ for the negative samples and we use L_2 distance for the distance metric d.

3.4 Association

Given the 3D representations that has been processed over space-time, the remaining task is to assign identities to each bounding box. We employ a simple association scheme that relies on the effectiveness of these representations. Specifically, the output vectors from the transformer $\widehat{\mathbf{h}}^{t,i}$ for all $i \in [P]$ at frame $t+1$ need to be associated to the existing tracklets up to t or to a new tracklet if no match exists. We use a simple tracking algorithm that takes one frame at a time and assigns track-ids based on the similarities between the past tracklets and the new detections.

First, we start with frame 1, and assign unique identities for each person. This initiates a set of tracklets $\mathcal{T} = \{\mathbf{Tr}_p\}$ for all person p in the first frame. Every tracklet is also associated with the corresponding human embeddings $\widehat{\mathbf{h}}^{1,k}$. Then we move on to frame 2 and calculate the affinity matrix between the detected humans $\mathcal{D} = \{\widehat{\mathbf{h}}^{2,i}\}$ in the frame 2, and all active tracklets $\mathcal{T} = \{\mathbf{Tr}_j\}$. We use Hungarian algorithm to assign detected humans to the tracklets, using the minimum distance between the last 20 embeddings vectors associated the tracklet and the embeddings vector of the detected human. This process continues for all frames sequentially. In this process, some of the tracklets might be missed and some detections might be not assigned. In some cases, there will be new humans added to the scene, and some people will leave or occluded from the scene. For the first case, we assign a new tracklet to the new person if there are no matches between old tracklets. For the latter case, we keep the old tracklets alive, even if there are no matches for few number of frames. We remove the tracklet if no new person is assigned to the tracklet continuously for 20 frames. A sketch of this tracking algorithm is shown in Algorithm 1.

4 Experiments

To evaluate the proposed approach, we first evaluate the efficacy of 3D representations over 2D representations, then provide ablations for the different components of our system and finally

Algorithm 1 Tracking Algorithm

procedure ONLINE TRACKING
Require: $\widehat{\mathbf{h}}^{t,i}$ for all $t \in [T], i \in [P]$, total number of frames T, maximum number of people per frame P.
  $\mathcal{T} \leftarrow \{\mathbf{Tr}_i(\widehat{\mathbf{h}}^{1,i}),\ \forall i \in [1, 2, ..., P]\}$ ▷ Initiate tracklets with detections from the first frame.
  for $t \in [2, 3, ..., T]$ **do**
    $\mathcal{D} \leftarrow \{\widehat{\mathbf{h}}^{t,i},\ \forall i \in [1, 2, ..., P]\}$ ▷ Create a detected human set.
    Calculate Affinity $\mathbb{A}_{i,j} \leftarrow \texttt{min}\{\tau, d(D_i, \mathbf{Tr}_j)\}$ for $D_i \in \mathcal{D}$ and $\mathbf{Tr}_j \in \mathcal{T}$
    $\mathcal{M}, \mathcal{T}_u, \mathcal{D}_u \leftarrow \texttt{Assignment}(\mathbb{A})$ ▷ Hungarian algorithm to assign detections to tracklets.
    $\mathcal{T} \leftarrow \{\mathbf{Tr}_j(\widehat{\mathbf{h}}^{t,i}),\ \mathbf{Tr}_j(age) = 0,\ \forall (i,j) \in \mathcal{M}\}$ ▷ assign detections to tracks.
    $\mathcal{T} \leftarrow \{\mathbf{Tr}_j(age) += 1,\ \forall (j) \in \mathcal{T}_u\}$ ▷ Increase the age of unmatched tracks by one.
    $\mathcal{T} \leftarrow \{\mathbf{Tr}_j(\widehat{\mathbf{h}}^{t,i}),\ \forall (i) \in \mathcal{D}_u,\ j = |\mathcal{T} + 1|\}$ ▷ Add new tracks for new detections.
    Kill the tracks with age ≥ 20.
  return Tracklets $\mathcal{T}$

compare against state-of-the-art tracking methods. We experiment on three datasets PoseTrack [1], MuPoTS [26], and AVA [12], which span a large variety of challenging sequences, including sports, daily activities and storytelling in movies. In all cases, we rely on detections from an off-the-shelf detection system [32], and we solve for the association, assigning identity labels to each bounding box using the proposed approach. As a result, the metrics we employ also focus on identity tracking on the bounding box level. Specifically, we report Identity switches (IDs), or the number of times a recovered trajectory switches from one ground truth tracklet to another. We also report the standard Multi-Object Accuracy (MOTA) [19] which summarizes errors from false positives, false negatives and ID switches. Finally, we also report the ID F1 score (IDF1) [33] which emphasizes the preservation of the track identity over the entire sequence.

Tracking people in edited media requires robustness against shot changes. For this evaluation we use AVA, which is a dataset of movies with actors annotated with identities. We use the test split from [28], which consists of sequences that contain shot changes. In this setting we *only* evaluate on tracklets that are persistent across shot changes, so to avoid being influenced by detection, we only report IDs and IDF1 on this dataset.

4.1 Implementation details

To train HMAR, we use a pretrained HMR model as the starting point. On top of the Resnet-50 backbone that HMR uses, we add additional layers from which we predict the texture flow to predict the texture map. We train this method using images from COCO [23], MPII [2] and Human3.6M [14]. We train the appearance head for roughly 500k iterations with a learning rate of 0.0001 and a batch size of 16 images while keeping the pose head frozen. Please see supplemental for more details. After training HMAR, we train an auto-encoder on the predicted texture map, which results in an encoded appearance embedding $\bar{\mathbf{a}}$. We train this using the same datasets as HMAR, and training lasts for about 100k iterations, with a learning rate of 0.0001. Finally, for training the transformer, we use the training set of PoseTrack [1] as this process requires videos with multiple people with labeled ids for training. Training lasts for 10k iterations with a learning rate of 0.001. All experiments are conducted on a single RTX 2080 Ti.

4.2 Quantitative evaluation

4.2.1 3D vs 2D

We first evaluate the effectiveness of 3D representations over 2D representations. To this end, we train a simpler version of our system that only uses one cue and compare with 2D and 3D versions of these cues. The full results of these experiments are presented in Table 1. We experiment with two popular 2D appearance features used in AlphaPose [8, 47] and Tracktor [4], which are convolutional features trained with Re-ID losses. We compare this against the proposed 3D appearance i.e. the encoded texture map. As we can see at the first three rows of Table 1, our 3D-based representation outperforms the popular 2D-based representations, except on AVA where we perform on par with Tracktor. Regarding the performance of pose/location cues, we compare 3D keypoints with the

corresponding 2D keypoints (coming from the projection of the 3D keypoints on the image plane). Again, the 3D-based cue outperform the 2D-based counterpart across the board.

	Embedding	2D/3D	Posetrack			MuPoTs			AVA	
			IDs↓	MOTA↑	IDF1↑	IDs↓	MOTA↑	IDF1↑	IDs↓	IDF1↑
App.	Re-ID (OSNet) [47]	2D	731	58.2	74.0	88	62.7	81.2	454	59.2
App.	Re-ID (Tracktor) [4]	2D	647	58.4	74.2	67	62.8	79.8	**242**	**64.6**
App.	Texture map (Ours)	**3D**	**567**	**58.6**	**74.4**	**44**	**62.9**	**82.2**	246	64.2
P&L	2D pose & location	2D	785	58.1	73.8	194	62.2	**80.8**	575	56.6
P&L	3D pose & location	**3D**	**566**	**58.6**	**74.3**	**52**	**62.8**	80.1	**280**	**64.2**

Table 1: *Effect of 3D Information:* To assess the effect of 3D information, we train our model using only one embedding that corresponds to either appearance or position/location and compare between the 2D and 3D representations. Using representations based on 3D information performs consistently better than the corresponding 2D-based representations.

4.2.2 Relative importance of different components

Having established the effectiveness of the 3D representation for tracking, we continue with a careful ablation of the components of our system. We report the performance of the pipeline by individually removing each of the 3D cues (i.e., appearance, pose and 3D location). We also remove the spatio-temporal aggregation of the transformer for ablation. The results for the different versions, as well as the full system are presented in Table 2.

These ablations show that 3D appearance is one of the most salient cue across the board. On PoseTrack and MuPoTs 3D location is more important than 3D pose (which is partially subsumed by location since that is based on keypoints). However on AVA, because of shot changes, the two most important cues are appearance and 3d pose, because across a shot boundary 3D pose is preserved while location is not. The transformer consistently adds to the overall robustness of the method. Note that PoseTrack training data is not very large (roughly 800 videos each less than a minute), particularly compared to the usual scale of the datasets that transformers are usually trained on, e.g. 300M images for JFT [7]. Potentially using larger training data could improve the performance even more.

Method	PoseTrack			MuPoTs			AVA	
	IDs↓	MOTA↑	IDF1↑	IDs↓	MOTA↑	IDF1↑	IDs↓	IDF1↑
w/o appearance	1314	56.9	72.1	73	62.7	80.7	689	54.3
w/o 3D pose	490	**58.8**	74.5	43	**62.9**	81.7	302	65.1
w/o 3D location	604	58.5	74.3	59	62.8	81.3	238	**66.2**
w/o transformer	511	57.8	74.5	43	**62.9**	82.6	252	65.5
Full system	**487**	**58.8**	**74.7**	**38**	**62.9**	**83.0**	**217**	65.0

Table 2: *Ablation of the components of our system:* We train our system by removing each cue at a time as well as the transformer for ablative study. We find that appearance is a salient cue across the board, 3D location is more important than 3D pose in PoseTrack and MuPoTs, but in AVA due to shot changes 3D pose (which is view invariant) is more useful than location. Transformer consistently improves the performance.

4.2.3 Comparison with the state-of-the-art

Lastly we compare our approach with the state-of-the-art tracking methods. For this comparison, we use four recent popular methods for tracking: Trackformer [27], Tracktor [4], Alphapose [8] and PoseFlow [40]. We run these methods out-of-the-box on all benchmarks and report their performance on identity tracking. For a more clear comparison, we provide results with two versions of our approach, one without using the transformer (Ours-v1), and one with the spatio-temporal aggregation of the transformer (Ours-v2). We do this, because our transformer is trained on PoseTrack, but some of the above baselines are not using PoseTrack for training [4, 27], so the Ours-v1 baseline is a more fair choice for a direct comparison. The full results are presented in Table 3. The two versions of our approach perform consistently better than previous work across the board. In Figure 6, we can also see qualitatively the benefits of our approach compared to these baselines. More specifically,

Figure 6: *Qualitative results.* We show samples from the three datasets we experiment on, PoseTrack [1], MuPoTs [26] and AVA [12]. We provide results for our method, AlphaPose [8] and Trackformer [27]. More video results are available in the Supplementary Material: `https://sites.google.com/view/t3po/`.

the design choices of our approach make it particularly suitable for challenging cases involving person-person occlusion, or shot changes which are common in edited media (AVA dataset). Please see our supplemental website for video results: `https://sites.google.com/view/t3po/`.

Method	Posetrack			MuPoTs			AVA	
	IDs↓	MOTA↑	IDF1↑	IDs↓	MOTA↑	IDF1↑	IDs↓	IDF1↑
Trackformer [27]	711	39.5	65.3	49	67.7	31.4	762	43.5
Tracktor [4]	535	49.9	67.8	51	78.3	**63.0**	346	52.9
Ours-v1	**511**	**57.8**	**74.7**	**43**	**82.6**	62.9	**252**	**65.5**
AlphaPose [8]	2534	46.2	71.4	108	74.1	46.1	944	44.4
FlowPose [40]	1220	22.4	67.6	52	71.7	28.4	333	58.3
Ours-v2	**487**	**58.8**	**74.5**	**38**	**83.0**	**62.9**	**217**	**65.0**

Table 3: *Comparison with state-of-the-art tracking methods:* We provide results for two versions of our approach. Ours-v1 does not include the transformer, while Ours-v2 applies the transformer for the spatio-temporal aggregation. Since our transformer is trained on PoseTrack, we provide these two versions for a more fair comparison to methods that do not train on this dataset (i.e., [4, 27]).

5 Concluding Remarks

We have demonstrated the value of using 3D representations for tracking people in a wide variety of videos. The proposed approach is simple and modular, and can easily incorporate extra representations such as face embeddings, which may be useful for movies. Our technique relies on having enough resolution on each person to enable lifting to 3D from 2D. Our approach is also susceptible to bad detections, since 3D features computed on non-human regions lead to spurious 3D representations. There are many applications of tracking people in video that have positive societal impact, such as medical diagnosis and rehabilitation, design of architectural spaces, scientific analysis of child development, home monitoring for elder care and more. However, such technology can also be misused for mass-surveillance.

References

[1] Mykhaylo Andriluka, Umar Iqbal, Eldar Insafutdinov, Leonid Pishchulin, Anton Milan, Juergen Gall, and Bernt Schiele. PoseTrack: A benchmark for human pose estimation and tracking. In *CVPR*, 2018.

[2] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2D human pose estimation: New benchmark and state of the art analysis. In *CVPR*, 2014.

[3] Anurag Arnab, Carl Doersch, and Andrew Zisserman. Exploiting temporal context for 3D human pose estimation in the wild. In *CVPR*, 2019.

[4] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *ICCV*, 2019.

[5] Christoph Bregler and Jitendra Malik. Tracking people with twists and exponential maps. In *CVPR*, 1998.

[6] Patrick Dendorfer, Aljosa Osep, Anton Milan, Konrad Schindler, Daniel Cremers, Ian Reid, Stefan Roth, and Laura Leal-Taixé. MOTChallenge: A benchmark for single-camera multiple target tracking. *IJCV*, 129(4):845–881, 2021.

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[8] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. RMPE: Regional multi-person pose estimation. In *ICCV*, 2017.

[9] David F. Fouhey, Weicheng Kuo, Alexei A. Efros, and Jitendra Malik. From lifestyle vlogs to everyday interactions. In *CVPR*, 2018.

[10] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. Recurrent network models for human dynamics. In *ICCV*, 2015.

[11] Rohit Girdhar, Georgia Gkioxari, Lorenzo Torresani, Manohar Paluri, and Du Tran. Detect-and-track: Efficient pose estimation in videos. In *CVPR*, 2018.

[12] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, Cordelia Schmid, and Jitendra Malik. AVA: A video dataset of spatio-temporally localized atomic visual actions. In *CVPR*, 2018.

[13] Peng Guan, Alexander Weiss, Alexandru O Balan, and Michael J Black. Estimating human shape and pose from a single image. In *ICCV*, 2009.

[14] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. *PAMI*, 36(7):1325–1339, 2013.

[15] Wen Jiang, Nikos Kolotouros, Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Coherent reconstruction of multiple humans from a single image. In *CVPR*, 2020.

[16] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *CVPR*, 2018.

[17] Angjoo Kanazawa, Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *ECCV*, 2018.

[18] Angjoo Kanazawa, Jason Y Zhang, Panna Felsen, and Jitendra Malik. Learning 3D human dynamics from video. In *CVPR*, 2019.

[19] Rangachar Kasturi, Dmitry Goldgof, Padmanabhan Soundararajan, Vasant Manohar, John Garofolo, Rachel Bowers, Matthew Boonstra, Valentina Korzhova, and Jing Zhang. Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol. *PAMI*, 31(2):319–336, 2009.

[20] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. VIBE: Video inference for human body pose and shape estimation. In *CVPR*, 2020.

[21] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3D human pose and shape via model-fitting in the loop. In *ICCV*, 2019.

[22] Oh-Hun Kwon, Julian Tanke, and Juergen Gall. Recursive bayesian filtering for multiple human pose tracking from multiple cameras. In *ACCV*, 2020.

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014.

[24] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. SMPL: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015.

[25] Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Mohamed Elgharib, Pascal Fua, Hans-Peter Seidel, Helge Rhodin, Gerard Pons-Moll, and Christian Theobalt. XNect: Real-time multi-person 3D motion capture with a single RGB camera. *ACM Transactions on Graphics (TOG)*, 39(4):82–1, 2020.

[26] Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Srinath Sridhar, Gerard Pons-Moll, and Christian Theobalt. Single-shot multi-person 3D pose estimation from monocular RGB. In *3DV*, 2018.

[27] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. *arXiv preprint arXiv:2101.02702*, 2021.

[28] Georgios Pavlakos, Jitendra Malik, and Angjoo Kanazawa. Human mesh recovery from multiple shots. *arXiv preprint arXiv:2012.09843*, 2020.

[29] Xue Bin Peng, Angjoo Kanazawa, Jitendra Malik, Pieter Abbeel, and Sergey Levine. Sfv: Reinforcement learning of physical skills from videos. 37(6), Nov. 2018.

[30] Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *PAMI*, 12(7):629–639, 1990.

[31] N Dinesh Reddy, Laurent Guigues, Leonid Pischulin, Jayan Eledath, and Srinivasa Narasimhan. TesseTrack: End-to-end learnable multi-person articulated 3D pose tracking. In *CVPR*, 2021.

[32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *PAMI*, 39(6):1137–1149, 2016.

[33] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *ECCV*, 2016.

[34] Michael Snower, Asim Kadav, Farley Lai, and Hans Peter Graf. 15 keypoints is all you need. In *CVPR*, 2020.

[35] Yu Sun, Qian Bao, Wu Liu, Yili Fu, Michael J Black, and Tao Mei. Monocular, one-stage, regression of multiple 3D people. *arXiv preprint arXiv:2008.12272*, 2020.

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[37] Jian Wang, Yunshan Zhong, Yachun Li, Chi Zhang, and Yichen Wei. Re-identification supervised texture generation. In *CVPR*, 2019.

[38] Xinshuo Weng, Yongxin Wang, Yunze Man, and Kris M Kitani. GNN3DMOT: Graph neural network for 3D multi-object tracking with 2D-3D multi-feature learning. In *CVPR*, 2020.

[39] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *ECCV*, 2018.

[40] Yuliang Xiu, Jiefeng Li, Haoyu Wang, Yinghong Fang, and Cewu Lu. Pose Flow: Efficient online pose tracking. *arXiv preprint arXiv:1802.00977*, 2018.

[41] Jiarui Xu, Yue Cao, Zheng Zhang, and Han Hu. Spatial-temporal relation networks for multi-object tracking. In *ICCV*, 2019.

[42] Xiangyu Xu, Hao Chen, Francesc Moreno-Noguer, Laszlo A Jeni, and Fernando De la Torre. 3D human pose, shape and texture from low-resolution images and videos. *PAMI*, 2021.

[43] Yihong Xu, Aljosa Osep, Yutong Ban, Radu Horaud, Laura Leal-Taixé, and Xavier Alameda-Pineda. How to train your deep multi-object tracker. In *CVPR*, 2020.

[44] Andrei Zanfir, Elisabeta Marinoiu, Mihai Zanfir, Alin-Ionut Popa, and Cristian Sminchisescu. Deep network for the integrated 3D sensing of multiple people in natural images. *NeurIPS*, 2018.

[45] Jason Y Zhang, Panna Felsen, Angjoo Kanazawa, and Jitendra Malik. Predicting 3D human dynamics from video. In *ICCV*, 2019.

[46] Yuxiang Zhang, Liang An, Tao Yu, Xiu Li, Kun Li, and Yebin Liu. 4D association graph for realtime multi-person motion capture using multiple video cameras. In *CVPR*, 2020.

[47] Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. Omni-scale feature learning for person re-identification. In *ICCV*, 2019.

[48] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. View synthesis by appearance flow. In *ECCV*, 2016.

[49] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *ECCV*, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] Upon publication we will release all of our code as an open source project.

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes]

(b) Did you mention the license of the assets? [N/A] We don't introduce new assets in this paper.

(c) Did you include any new assets either in the supplemental material or as a URL? [N/A]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]