MULTI-VIEW HUMAN TRACKING AND 3D LOCALIZATION IN RETAIL

Akash Jadhav

Noque.store, India

ABSTRACT

In recent years, retail stores have seen traction in bringing online shopping experience to offline stores via autonomous checkouts. Autonomous checkouts is a computer vision-based technology that needs to understand three human elements within the store: who, where, and doing what. This paper addresses two of the three elements: who and where. It presents an approach to track and localize humans in a multi-view camera system. Traditional methods have limitations as they: (1) fail to overcome substantial occlusion of humans; (2) suffer a lengthy processing time; (3) require a planar homography constraint between camera frames; (4) suffer swapping of labels assigned to a human. The proposed method in this paper handles all the aforementioned limitations. The key idea is to use a hierarchical association model for tracking, which uses each human's clothing features, human pose orientation, and relative depth of joints, and runs at over 23fps.

KEYWORDS

Multi-view, Data Association, Tracking, Localization.

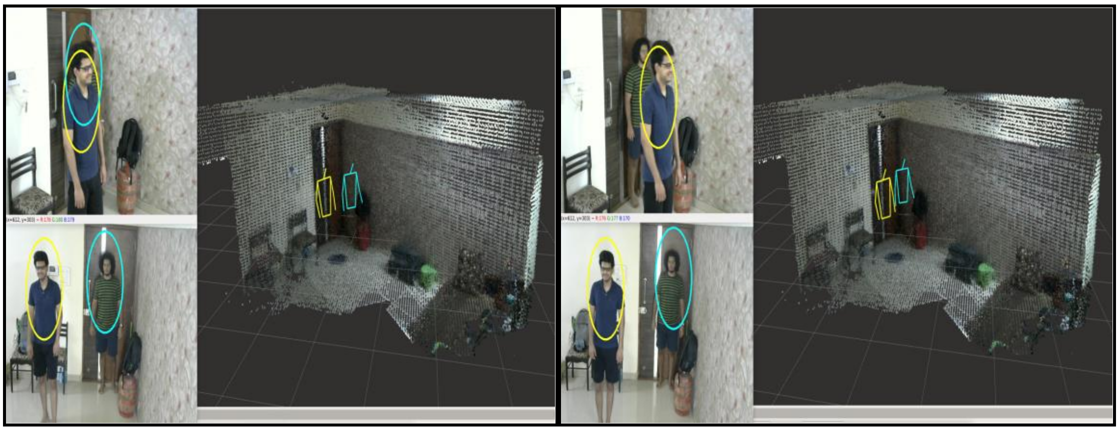
1. INTRODUCTION

There has been significant research work done in estimating and tracking the pose of a human from a single view [4, 2, 16] and multi-view images [5, 12, 17, 18], yet few existing methods have been crafted to tackle the problem of tracking and localizing humans in a retail store. Problems faced in tracking and localizing humans in the retail stores are especially challenging for computer vision algorithms due to: significant heavy occlusion of humans, similarly dressed humans, swapping of labels assigned to humans, and having an extensive baseline between cameras. However, the potential application of estimating and tracking the 3D pose of humans is in retail stores. It provides an autonomous checkout experience to the customer and gathers a lot of analytics, which helps increase the sales and helps understand customer behavior, which brings the potential of online retail to offline retail.

Estimating 2D poses of humans in the images is a well-researched problem in deep learning [1, 9]. With further optimization [26] of these deep learning models, they can run in real-time [29]. The task of tracking humans in a single view and multi-view camera system [8, 11, 5], further combining the 2D poses from multiple perspectives to generate 3D skeletons, has also been explored [12]. However, none of these methods work robustly for the application of tracking and localizing humans in retail stores due to: (1) substantial occlusion of humans; (2) lengthy processing time; (3) requirement of a planar homography constraint between camera frames; (4) swapping of label assigned to a human. This paper proposes a hierarchical association model to find correspondences between 2D poses in multiple views, employing them to generate and track 3D skeletons. By maintaining the exact label of a human throughout the journey in the 3D space,

David C. Wyld et al. (Eds): ICAIT, CBIoT, WiMo, CRYPIS, ICDIPV, CAIML, NLCA - 2022
pp. 181-192, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.121214

the resulting method provides a significant improvement over previous methods and corrects the errors associated with multi-view human tracking. Example results can be seen in Figure 1.



(a) (b)

Figure 1. Results of the proposed method. The two images on the left of (a) and (b) are from the dual-view camera system, which shows humans detected with a particular color bubble (label).

The right of (a) and (b) shows the localized 3D skeletons in the 3D space with the same color. (a) Results show tracking and localization of humans in an occluded scene, where the blue bubble is the predicted state of the occuluded human in the upper view; (b) Results show tracking and localization of humans in case of missed detection, where the blue 3D skeleton is the predicted state of the occuluded human with missed detection in the upper view.

This association model helps in estimating the 3D skeleton in 3D space for a prolonged time under substantial occlusions and missed detections of humans in a multi-view system, which enables to understand some complex actions like picking an object, placing the object back on the rack, dropping the object in the bag or cart, passing or throwing an object to another human, and catching an object by another human.

The proposed algorithm is experimented on the generated dataset [31], which is generated using two Logitech C270 cameras with two-to-three people maneuvering in the surrounding. This dataset consists of a readme.pdf file that defines the content in the dataset. Unfortunately, there is no similar dataset available that includes rigid camera poses in 3D space with a fixed number of humans maneuvering in the same 3D space. The following section provides litrature for some previous approaches and comapers them to solving this problem to the proposed algorithm.

2. Literature Review

Estimation of 2D human poses from a monocular image can be categorized into two categories: (1) single-person 2D pose detection [20, 22, 23]; (2) multi-person 2D pose detection [1,9, 21]. Toshev and Szegedy [22] provide a regressor to directly estimate the 2D joint coordinates in an image. An end-to-end deep learning model that learns spatial models for 2D pose estimation was presented in [23]. Deep convolutional network-based pose estimators result in a significant increase in accuracy and provide a basis for more difficult pose estimation tasks such as multi-person 2D pose estimation [1, 9]. Cao et al. [1] presents a fusion of joint confidence map and a learned vector that defines the relationship between the joints, and estimates the 2D poses.

Some researchers have proposed a single-view and multi-view human detection and tracking algorithm to overcome the limitations of humans being occluded in the camera scene. Cai and Aggarwal [10] extend a single-camera tracking system by switching to other views when the system predicts that the current camera will no longer have a clear view of the object. Krumm et al. [13] combine information from multiple stereo cameras in the 3D space. They perform background subtraction and then detect human-shaped blobs in 3D space. For detecting and tracking the same human in numerous images, each person is assigned a color histogram. Back-projection in 3D space estimates the 3D points guaranteed to lie inside the detected objects. Wojke et al. [8] tries to detect and track objects in a single view by extracting a feature vector from the bounding box assigned to the detected objects and compares this feature vector with other feature vectors. Even though this method attempts to resolve occlusions, the underlying problem of using such features is that the overlapping bounding boxes might get corrupted, as shown in Figure 2 (a). In the approach proposed in this paper, a bounding box is computed using the 2D joint coordinates to extract the clothing features, as shown in Figure 2 (b).

Khan et al. [5] requires a planar homography constraint between multiple cameras to track multiple humans. This constraint creates a dependency that the cameras need to see the floor with the human feet pixels always visible to project them in all other frames. Then a clustering algorithm associates these projected pixels to a particular human. The floor and human feet would not always be visible from the camera feed in a retail store as shown in Figure 2. Bridgeman et al. [3] provides a method which only depends on 2D joint association without any appearance features to track and generate a 3D pose. This approach results in the swapping of labels when two humans pass close by with the same orientation.

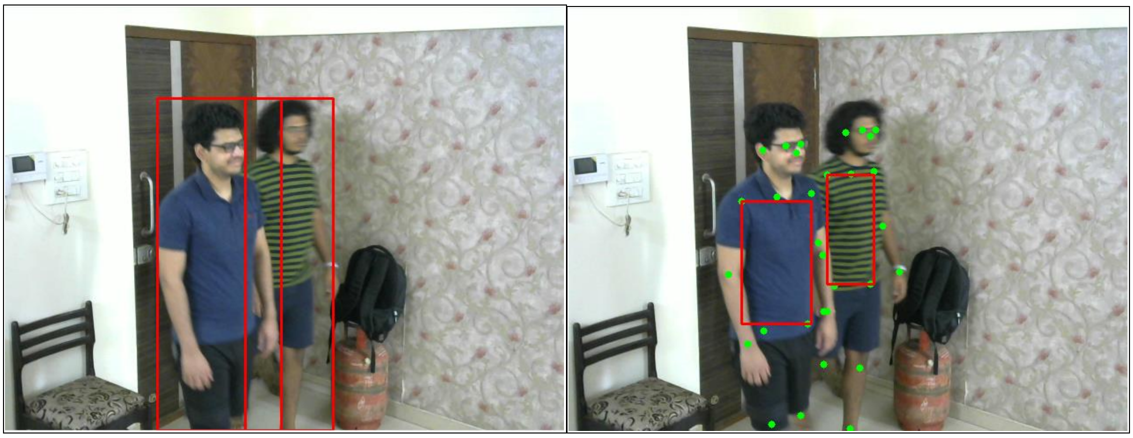

(a) (b)

Figure 2. In computer vision, estimatimating the six degree-of-freedom camera pose in the world frame from n 3D-to-2D point correspondences is a fundamental and well understood problem. This pose could be estimated with minimum 6 correspondences in 3D sapce and image pixels, using the well known Direct Linear Transform (DLT) algorithm. To improve the accuracy of the DLT, Perspective-n-Point with Ransac is used [27]. In the proposed framework, to estimate the orientation of human in the image, rough estimates of the 3D coordinates are taken for the 2D joint pixels. For rough estimates of the 3D coordinates of an object and its corresponding 2D joint pixels, a scaled translation of object origin is obtained, but the orientation remains within limits of 3 to 5 degrees across all axis.

Depth is an useful representation for actions in the physical environments. Monocular depth estimation remains a challenging problem that is heavily underconstrained. To solve it, one must exploit many visual cues, as well as long-range context and prior knowledge. This calls for learning-based techniques [6, 7, 19]. Ranftl et al. [19] proposes a robust training objective that is invariant to changes in depth range and scale, uses principled multi-objective learning approach to combine data from different sources, and highlights the importance of pretraining encoders on auxiliary tasks. Ranftl et al. [19] provides a pretrained model [28] which estimates the relative depth information from a monocular image and runs in real time. In the proposed framework, relative depth information of neck joint is used for the betterment in data association when the clothing features and human orientations are approximately same.

The scene with a higher number of occluded objects would be challenging to resolve for any of the previous methods. Not only are there cases of near-total occlusion, but similarly dressed people would also be a challenge. Using just the color distributions, or full human bounding box features, or 2D joint coordinates for region matching across cameras would lead to the incorrect association and result in swapping of labels or assigning a new label to the human. A hierarchical association model is proposed in the approach in order to overcome these limitations, which uses clothing features, human orientation data which is computed using the 2D joint pixels [27], and relative depth estimate [19,28] of the neck joint.

With the advancement in deep convolutional neural networks, it is easy to extract many features from an object. How well these features are associated with objects across all the frames defines a valuable tracking system. Feature vectors are computed from the bounding boxes of detected objects using similar convolutional neural networks. Wojke et al. [8] employs a feature extraction model from bounding boxes as shown in Figure 2(a), trained on a reidentification dataset [24]. This dataset contains over 1,100,000 images of 1,261 pedestrians, making it well suited for deep metric learning in a people tracking context. The method proposed in this paper employs the bounding box computed from the 2D joints detected as shown in Figure 2 (b), hence a clothing feature extraction model is trained on the reidentification dataset [24].

Single view and multi-view tracking algorithms adopt a single conventional hypothesis tracking methodology with recursive Kalman filtering and frame-by-frame data association. In the approach proposed in this paper, Kalman filter-based tracking is employed in both image coordinates and 3D space coordinates. In image coordinates, the bounding boxs, as shown in Figure 2 (b), are tracked, and in 3D space coordinates, the 3D joints of the skeleton are tracked with Kalman Filter. Tracking in both image coordinates and 3D coordinates provides robust and continuous 3D pose estimation in the 3D space. This technique enables the possibility to understand some complex actions performed by the 3D skeleton of humans in the 3D space.

3. METHODOLOGY

The proposed framework takes as input images from multiple cameras, camera calibration parameters, and the rigid pose of cameras in the 3D space. The multi-view images are passed through a pose detector [29] and a monocular depth estimator [19, 28], providing 2D pose estimations and relative depth information in each frame. Two successive processes are applied to the 2D pose data: the first step computes the bounding box from the 2D poses, and the second step computes the orientation of humans from their 2D poses in the images. From the bounding boxes, clothing features are computed using a feature extracting neural network. The relative depth of neck joints is estimated in each image using the relative depth information and 2D joint pixels. Further, the relative depth of neck joints, human orientation data, and the human clothing features are fed as input to the hierarchical data association model, which assigns a label to every human, ensuring consistency between multiple views. This set of labels with humans is used to track bounding boxes in the images and 3D joints in the 3D space. 3D joints are estimated if the human 2D joints are seen in two or more views using robust triangulation. In each image, bounding boxes are tracked instead of 2D joints to reduce the time complexity during runtime. Tracking of the bounding boxes ensures the tracking of the 2D joints, as mapping is stored between the 2D joints and the bounding box. If 2D pose measurements are available for a label assigned to a human, they are passed to the 3D joints estimation and tracking block. If the 2D pose measurements are not available, the predicted state of 2D joints from the tracked bounding boxes is passed to the 3D joints estimation and tracking block. A system overview is presented in Figure 3.

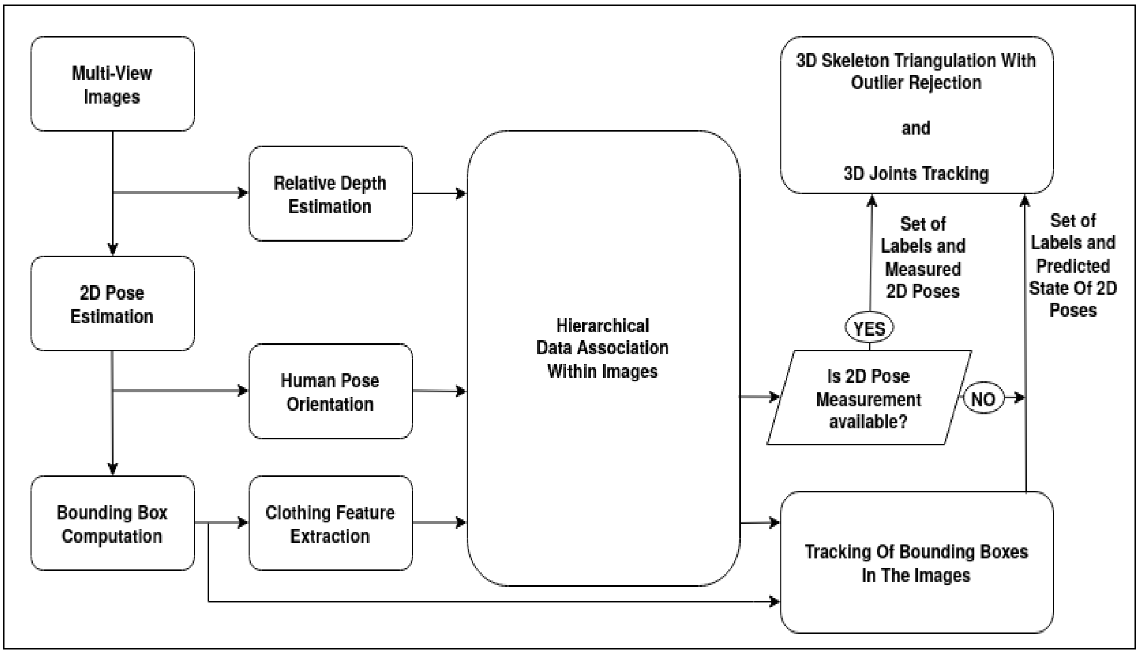

Multi-View Images
Relative Depth Estimation
2D Pose Estimation
Human Pose Orientation
Bounding Box Computation
Clothing Feature Extraction
Hierarchical Data Association Within Images
3D Skeleton Triangulation With Outlier Rejection
and
3D Joints Tracking
Set of Labels and Measured 2D Poses
Set of Labels and Predicted State Of 2D Poses
YES
Is 2D Pose Measurement available?
NO
Tracking Of Bounding Boxes In The Images

Figure 3. An overview of the pipeline.

The pose detector [29] estimates a total of 18 joints on a human in an image. To avoid the corruption in features due to overlapping bounding boxes, as shown in Figure 2 (a), a bounding box is computed using the left shoulder joint, right shoulder joint, right hip joint, and left hip joint as shown in figure 2 (b). This bounding box is passed to two functional blocks of the pipeline: (1) a feature extraction block to compute a feature vector from the clothing; (2) the bounding box tracking block.

Once the 2D pose of a human is detected, six joints are used to compute the orientation 27] of human in an image: neck joint, left shoulder joint, right shoulder joint, right hip joint, left hip joint, and the joint between the right hip joint and the left hip joint. These six joints always lie on a plane in the 3D space, the distances between these six joints always remain the same, and a reference frame could be attached to one of these joints. The 3D coordinates used for the neck joint, left shoulder joint, right should joint, right hip joint, and left hip joint are (0.0, 0.0, 0.0), (-250.0, 0.0, 0.0), (250.0, 0.0, 0.0), (250.0, -500.0, 0.0), (-250.0, -500.0, 0.0), and(0.0, -500.0, 0.0), where the readings are in milli-meters, and the neck joint is considered to be the origin. Relation between image pixels (u,v), their corresponding 3D points (X, Y, Z), intrinsic camera matrix (K), and the pose (rotation (R) and translation (T)) is given in Eq. 1, where K and RT are of size 3x3 and 3x4 respectively.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} r11 & r12 & r13 & t1 \\ r21 & r22 & r23 & t2 \\ r31 & r32 & r33 & t3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \qquad (1)$$

To extract a feature vector from the bounding box a convolutional neural network has been trained on a re-identification dataset [24]. This dataset contains over 1,100,000 images of 1,261 pedestrians, making it well suited for clothing feature extraction in a people tracking context. The convolutional neural network architecture of my network is shown in Table 1. Table 1 (a) displays the architecture of the encoder network. It consists of 5 layers with 5x5 filters and a stride of 2 using ReLU as an activation function. The output size from each layer should be read as height x width x channels. Table 1 (b) displays how the last activation map from the encoder network is transformed into the latent vector z, where the latent dimension is set to 10. Table 1 (c) displays how the output vector of size 1x768 is transformed into a tensor of shape 3x2x128. Note that the network is symmetrical with the equal number of layers, filter size, stride length, and activation as the encoder network. The training process of this network architecture is out of scope for this paper. One forward pass of 32 bounding boxes takes approximately 30ms on an Nvidia GeForce GTX 1070 GPU. Thus, this network is well suited for online tracking.

Table 1. The network layers and their respective output sizes are shown. The terms "C", "S" and "F" stand for channels, stride and filter size respectively. (a) Encoder Network architecture displaying activation function, number of channels, stride and filter for each layer in the encoder network. (b) Latent space architecture displaying the flow of data received from the encoder. Note that z-mean and z-std are separate fully connected layers. They are placed in the same row since they are computed in parallel. (c) Decoder Network architecture displaying activation function, number of channels, stride and filter for each layer in the decoder network.

Encoder Layers	**Output Size**
Input image	96x64x3
Conv1 - ReLU, C: 8, S: 2x2, F: 5x5x3	48x32x8
Conv2 - ReLU, C: 16, S: 2x2, F: 5x5x8	24x16x16
Conv3 - ReLU, C: 32, S: 2x2, F: 5x5x16	12x8x32
Conv4 - ReLU, C: 64, S: 2x2, F: 5x5x32	6x4x64
Conv5 - ReLU, C: 128, S: 2x2, F: 5x5x64	3x2x128
Reshape - Flatten	1x768

(a)

Latent Space Layers	**Output Size**
Flattened vector	1x768
Fully connected z-mean and z-std	1x10 + 1x10
Generated latent vector	1x10
Fully connected - ReLU	1x768

(b)

Decoder Layers	**Output Size**
Activation:ReLU, Fully-connected	1x768
Reshape - Tensor form	3x2x128
Deconv1 - ReLU, C: 64, S: 2x2, F: 5x5x128	6x4x64
Deconv2 - ReLU, C: 32, S: 2x2, F: 5x5x64	12x8x32
Deconv3 - ReLU, C: 16, S: 2x2, F: 5x5x32	24x16x16
Deconv4 - ReLU, C: 8, S: 2x2, F: 5x5x16	48x32x8
Deconv5 - ReLU, C: 3, S: 2x2, F: 5x5x8	96x64x3

(c)

From the relative depth information obtained from the mono-depth estimation model [19, 28] and 2D poses obtained from the pose detection model [29], humans are ranked according to their relative depth of the neck joint in the image. Later, for these relative depth points, a scaled distance is assigned as per their rank to create a scaled 3D coordinate. Finally, to associate humans to a particular label, these 3D coordinates are transformed onto other camera frames and checked with the neck's rank of relative depth point in the other camera frames. Refer to Figure 4 for reference; in view 1, the rank of cylindrical objects with respect to their relative depth points are [P1, P2]. A scaled distance 1D1 and 1D2 (1D1<1D2) is assigned for P1 and P2, in view 1. When these depth points are transformed to view 2, 2D1 and 2D2 are obtained (2D1<2D2). In view 2, the rank of cylindrical objects with respect to their relative depth points are [P1, P2], so 2D1 is associated with P1, and 2D2 is associated with P2.

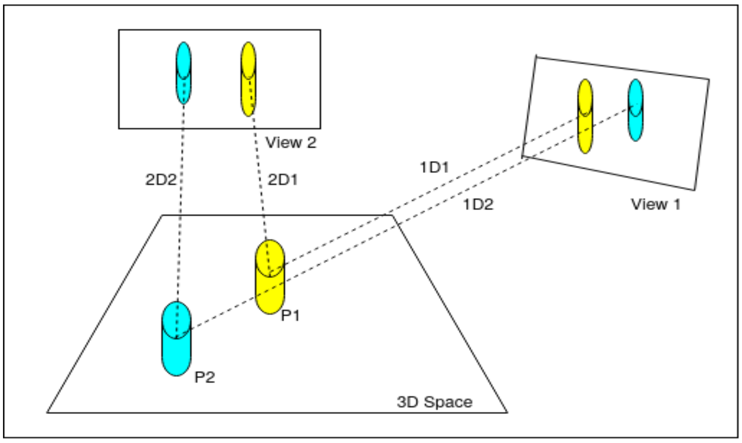


Figure 4. The figure shows two cylinderical objects standing in the 2D space, and the scene being viewed by two cameras.

The Hierarchical data association model takes as input a human vector which consists of: (1) the extracted clothing features; (2) human pose orientation; (3) relative depth of the neck joint. The hierarchical data association model computes an association mapping of labels with 2D poses and association mapping of labels with bounding boxes. As the N number of people inside the store are known, first the human data vectors are clustered into N labels using the clothing feature vector and human orientation data. This clustering is done by comapring the clothing features and human orientations data in the same image and later with the clothing features and human orientations data in every other image. The clothing appearance features are associated using a metric which computes a scalar value by comparing two feature vectors of size 1x768. This metric measures the cosine distance d between two normalized clothing feature vectors v1 and v2 as shown in Eq. 2. Again, a binary variable is introduced as shown in Eq. 3, to indicate if an association is admissible according to the above metric, where b is the binary value and t is a suitable threshold computed on the dataset [24].

$$d(v1, v2) = (1 - v2.T * v1) \tag{2}$$

$$b(v1,v2) = 1[d(v1,v2) \leq t] \tag{3}$$

If N different clusters are not formed by associating just the clothing features, the human orientations which are obtained with respect to the camera coordinates are transformed to the 3D space coordinates and compared with the other available human orientations. A binary variable, similar in Eq. 3 is used to compare if the difference between two human orientations is within a threshold. If still N different clusters are not formed by associating the clothing features and human orientations, humans are orientated using the relative depth information of neck joints. Just the relative depth information alone is not used to associate humans to labels, as this information is usable only upto a particular distance. Once N different clusters are formed, the multi-view tracking system is initialized and creates a set of labels and human vectors. Later, this set of labels and human vectors is used to associate human vectors from the multi-view system to a particular label. If a human assigned to a particular label is seen in two or more views, a 3D skeleton is estimated using robust triangulation. The 3D location of each joint s_{Ij} in a skeleton with label I is optimised using Eq. 4.

$$\underset{s_{Ij}}{argmin} \sum_{c} \sum_{I} alpha_{ij}{}^{c} \| P_c(s_{Ij} - p_{ij}{}^{c}) \|, \{p_i \in I, c \in C\} \tag{4}$$

where, $P_c(s_{Ij})$ is the projection of s_{Ij} in a camera c. This results in a set of 3D skeletons per frame, and RANSAC is used to eliminate the outlier pose detections. The bone lengths of the resultant skeleton are thresholded to remove any remaining outlier 3D joints.

To improve the localization of joints in the 3D space, the proposed framework implements Kalman filter-based tracking of bounding boxes in each image and 3D joints. The 2D joints are mapped with the bounding boxes. As the bounding boxes are tracked, the 2D joints get tracked by default. Tracking of bounding boxes is done instead of 2D joints to reduce the time complexity of the pipeline during runtime. If the 2D pose measurements are available, they are used to compute and track the 3D joints. If the 2D pose measurements are not available, the 2D joints obtained from the predicted state of bounding boxes compute and track the 3D joints. Kalman filter is applied in tracking to smoothen the final results, to compensate for the missed detections, and substantial occlusions of humans for a few frames.

4. EVALUATION

The method proposed in this paper is evaluated on the generated dataset [31], which consists of two cameras with two-to-three people maneuvering in the surrounding. Limitations of some previous methods [8, 5] on this dataset are described in this section. The specification of the dataset used for evaluation is shown in Table 2.

Table 2. Properties of the datasets used for qualitative evaluation: number of cameras (C); number of people (P); camera resolution (R);and number of frames (F).

Dataset [31]	C	P	R	F
2person.zip	2	2	480p	130
3person.zip	2	3	480p	330

The proposed method's feature extraction process is compared with the state of the art tracking method's [8] feature extraction process. The feature extraction model of the proposed method and the state-of-the-art method [8] is tested on Figure 2 (b) and Figure 2 (a) respectively. The comparison is made by calculating the cosine distance(CD) from Equation 2 between two human's bounding box feature vectors in each image, where, Lower cosine distance indicates more similarity between the two feature vectors.. In Figure 2 (a), the bounding boxes are generated by detecting humans using the yolov4 deep learning model [25], and in Figure 2 (b), the bounding boxes are generated by the 2D joints detected [29]. The cosine distances computed between two humans in Figure 2 (a) and Figure 2 (b) are shown in Table 3. The cosine distance between two humans in Figure 2 (a) is less than that in Figure 2 (b) due to the overlapping of bounding boxes which leads to corruption in the feature vectors. When the number of humans maneuvering in the scene increases, there would be more overlapping in bounding boxes, adding errors in the human-label association process.

Table 3. Metric values between two feature vectors.

Figure	CD between two clothing feature vectors
Figure 2 (a)	0.3489
Figure 2 (b)	0.9427

To assess the quality of the proposed system, the number of label switches are computed, a metric commonly used in multi-object tracking [15] that counts the number of times a tracked object is assigned a new identity. The results are shown in Table 4. Previous method [8, 30] which tracks humans in a single view using just the bounding box features results assigning new labels to humans under substantial occlusion, as shown in Figure 5 (a). Wojke et al. [8] fails to maintain the label assigned initially to a human throughout the journey in the 3D space. It fails to track a human in a single view and cannot be scaled to a multi-view tracking system. The method [5] does not work on the provided dataset [31] as there is no planar homography constraint between the camera frames, and in a retail store, it is tough to make every camera view the floor.

Table 4. The number of frames (F), tracked people (TP), and Label switches (LS) for out and previous methods [8, 5] in each dataset.

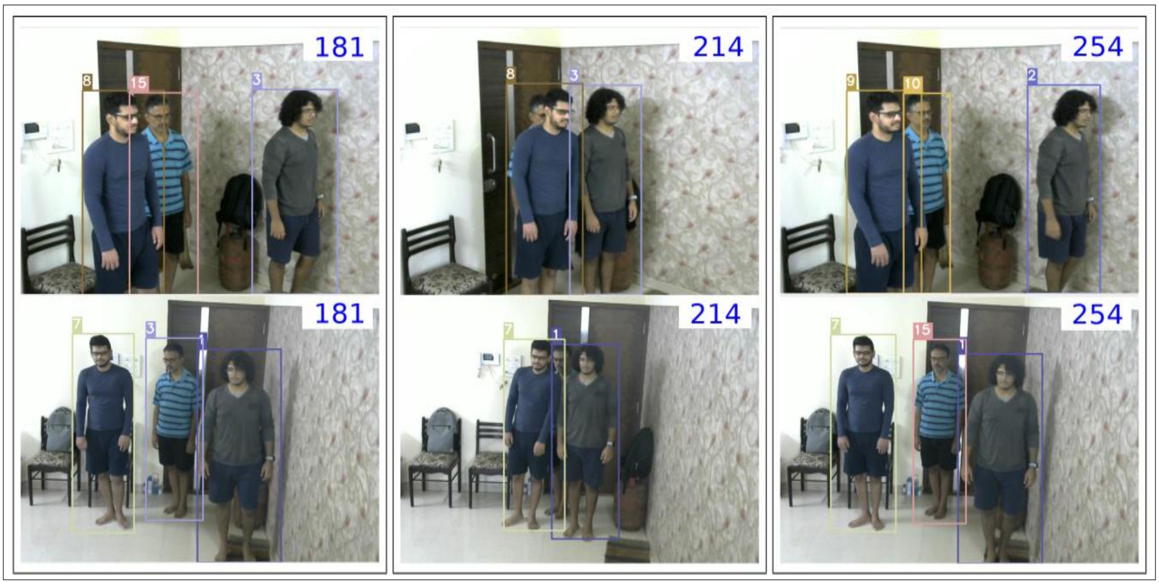
Dataset [31]	F	TP	LS - Ours	LS - [8]	LS - [5]
2person.zip	130	2	0	8	N.A.
3person.zip	330	3	0	21	N.A.

In the 2person.zip and 3person.zip dataset, all humans maintained their label for the duration of the sequence, even during close contact, substantial occlusion, and missed detections, which are observable in Figure 5 (b). The proposed method in this paper succeeds in maintaining the label assigned initially to a human throughout the journey in the 3D space. It also succeeds in tracking a human in a single view and a multi-view system.

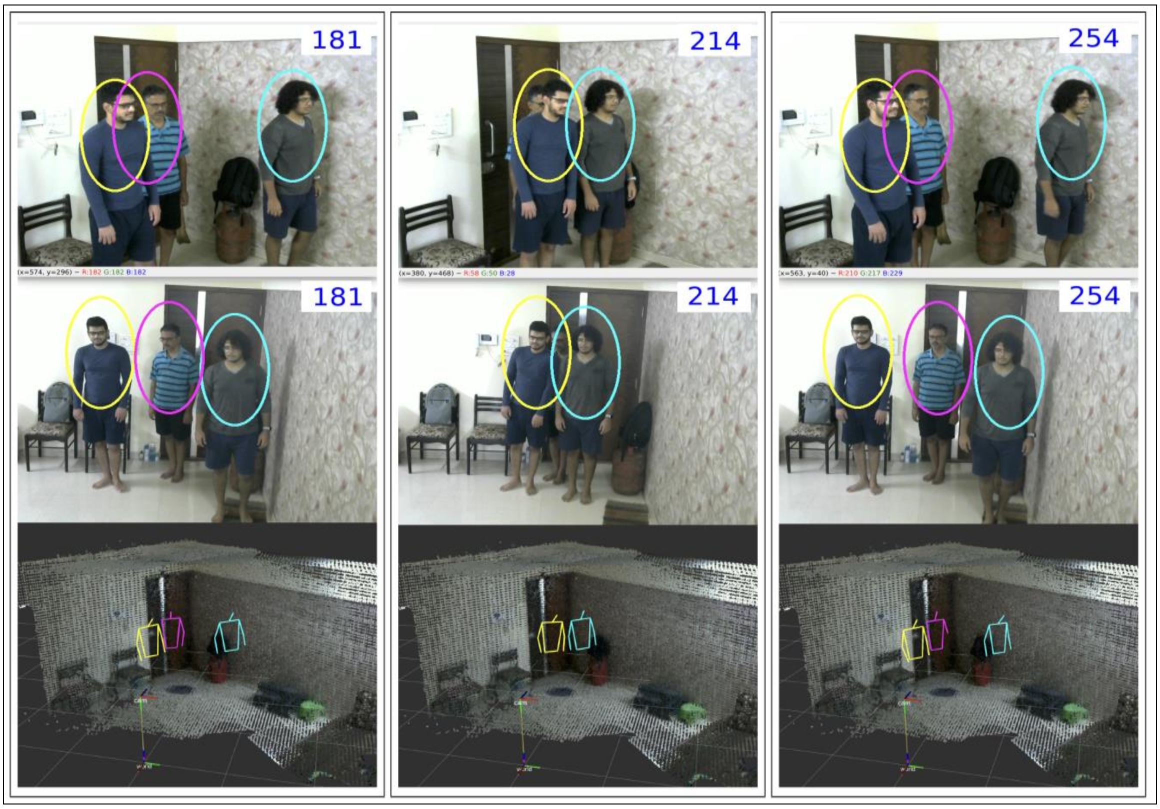
The method is tested on a system with an Intel i7 2.2GHz processor. 16GB of RAM, and 12GB Nvidia GeForce GTX 1070 GPU. The deep neural networks run over GPU and tracking algorithms run over CPU. The parallelized implementation runs at over 23fps on the dataset [31]. The 2Dpose association stage is the most computationally expensive, and the time taken to track the 2D bounding boxes and 3D skeletons adds a little latency. The methods which use pictorial structure models [12, 14] for association, run at approximately 1fps and 10fps respectively.

5. CONCLUSIONS

This paper presents a new method for computing and tracking 3D skeleton of humans in a multi-view camera system. The proposed hierarchical model for associating humans to labels compensates for errors in overcoming substantial occlusions of humans, does not have any constraints like consistency in planar homography between the cameras, and can identify correspondences between humans in different viewpoints. Moreover, the algorithm is capable of running at over 23fps, and tracks the 3D skeletons for a prolonged time. In future, the association algorithm could be made more robust by researching some deeplearning models which predict the human skeleton depth from monocular cameras instead of the relative depth which is currently used in the proposed approach. Tracking of 3D skeletons for a prolonged time enables to understand better some complex actions done by humans in the retail store.



(a)



(b)

Figure 5. Left images show results on frame before occlusion, center images show results on frame during occlusion, and right images show results on frame after occlusion. (a) Results on Wojke et al. [8, 30] method. (b) Results on the proposed method.

REFERENCES

[1] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, (2017) “Realtime multi-person 2d pose estimation using part affinity fields”, IEEE CVPR.

[2] M. Andriluka, S. Roth, and B. Schiele, (2010) “Monocular 3d poseestimation and tracking by detection”, IEEE CVPR.

[3] Bridgeman, Lewis and Volino, Marco and Guillemaut, Jean-Yves and Hilton, Adrian, (2019) “Multi-Person 3D Pose Estimation and Tracking in Sports”, IEEE CVPR (Workshops).

[4] Rochette, Guillaume and Russell, Chris and Bowden, Richard, (2019) “Supervised 3D PoseEstimation from a Single Image using Multi-View Consistency”, BMVC.

[5] Khan S.M., Shah M., (2006) “A Multiview Approach to Tracking People in Crowded Scenes Using a Planar Homography Constraint”, ECCV.

[6] D. Hoiem, A. A. Efros, and M. Hebert, (2005) “Automatic photo pop-up”, ACM Transactions on Graphics.

[7] A. Saxena, M. Sun, and A. Y. Ng., (2009) “Make3D: Learning 3D scene structure from a single still image”, IEEE PAMI.

[8] Wojke, Nicolai and Bewley, Alex and Paulus, Dietrich, (2017) “Simple Online and Realtime Tracking with a Deep Association Metric”, IEEE International Conference on Image Processing.

[9] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, andB. Schiele, (2016) “Deepercut: A deeper, stronger, and faster multi-person pose estimation model”, ECCV.

[10] Cai, Q. and Aggarwal, (1998) “Automatic tracking of human motion in indoorscenes across multiple synchronized video streams”, ICCV.

[11] Wojke, Nicolai and Bewley, Alex, (2018) "Deep Cosine Metric Learning for Person Re-identification", IEEE Winter Conference on Applications of Computer Vision (WACV).

[12] V. Belagiannis, S. Amin, M. Andriluka, B. Schiele, N. Navab, and S. Ilic. (2016) "3d pictorial structures revisited: Mul-tiple human pose estimation", IEEE PAMI.

[13] Krumm, J., Harris, S., Meyers, B., Brumitt, B., Hale, M., and Shafer, S., (2000) "Multi-camera multi-person tracking for easy living", IEEE International Workshopon Visual Surveillance.

[14] J. Dong, W. Jiang, Q. Huang, H. Bao, and X. Zhou, (2019) "Fast androbust multi-person 3d pose estimation from multiple views", IEEE CVPR.

[15] A. Milan, L. Leal-Taix, I. D. Reid, S. Roth, and K. Schindler, (2016) "Mot16: A benchmark for multi-object tracking", CoRR.

[16] D. Tome, C. Russell, and L. Agapito, (2017) "Lifting from thedeep: Convolutional 3d pose estimation from a single image", IEEE CVPR.

[17] H. Joo, T. Simon, X. Li, H. Liu, L. Tan, L. Gui, S. Baner-jee, T. S. Godisart, B. Nabbe, I. Matthews, T. Kanade,S. Nobuhara, and Y. Sheikh, (2017) "Panoptic studio: A massively multiview system for social interaction capture", IEEE Trans-actions on Pattern Analysis and Machine Intelligence.

[18] L. Sigal, M. Isard, H. W. Haussecker, and M. J. Black, (2011) "Loose-limbed people: Estimating 3d human pose and mo-tion using non-parametric belief propagation", international Journal of Computer Vision.

[19] Ranftl and Katrin Lasinger and David Hafner and Konrad Schindler and Vladlen Koltun, (2020) "Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer", IEEE PAMI.

[20] P. F. Felzenszwalb and D. P. Huttenlocher, (2004) "Pictorial structures for object recognition", International Journal of Computer Vision.

[21] M. Kocabas, S. Karagoz, and E. Akbas, (2018) "Multiposenet: Fastmulti-person pose estimation using pose residual network", ECCV.

[22] A. Toshev and C. Szegedy, (2014) "Deeppose: Human pose esti-mation via deep neural networks", IEEE CVPR.

[23] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, (2016) "Convolutional pose machines", IEEE CVPR.

[24] L. Zheng, Z. Bie, Y. Sun, J. Wang, C. Su, S. Wang, and Q. Tian, (2016) "MARS: A Video Benchmark for Large-Scale Person Re-Identification", ECCV.

[25] Bochkovskiy, Alexey and Wang, Chien-Yao and Liao, Hong-Yuan Mark, (2020) "YOLOv4: Optimal Speed and Accuracy of Object Detection", IEEE CVPR.

[26] Nvidia: Torch2TRT,
https://nvidia-ai-iot.github.io/torch2trt/v0.2.0/index.html

[27] Perspective-n-Point, Wikipedia,
https://en.wikipedia.org/wiki/Perspective-n-Point

[28] MIDAS v21 Small Model,
https://github.com/AlexeyAB/MiDaS/releases/download/midas_dpt/midas_v21_small-70d6b9c8.pt/

[29] Nvidia: PoseTRT,
https://github.com/NVIDIA-AI-IOT/trt_pose

[30] Wojke et al. [8] Implementation,
https://github.com/nwojke/deep_sort

[31] Dataset,
https://drive.google.com/file/d/11OqvWwXXqnR8KP_QmVk07Pgwg20rw2H3/view?usp=sharing

AUTHORS

Akash Jadhav
Founder, Noque.store
akash.jadhav@noque.store
https://www.linkedin.com/in/akash-jadhav-2201/