3D Human Pose Estimation From Multi Person Stereo 360° Scenes

M. Shere, H. Kim and A. Hilton
Centre for Vision, Speech and Signal Processing
University of Surrey, UK
{m.shere, h.kim, a.hilton}@surrey.ac.uk

Abstract

This paper presents a human tracking and 3D pose estimation algorithm for use with a pair of 360° *cameras. We identify and track an individual throughout complex, multi-person scenes in both indoor and outdoor environments using appearance models and positional data, and produce a temporally consistent 3D skeleton by optimising a skeleton of realistic joint lengths over joint positions produce by Convolutional Pose Machines (CPMs). Our results show an average improvement of 22.67% over state of the art deep learning approaches for tracking, as well as reasonable estimates for pose using just two cameras.*

1. Introduction

Human motion capture has become an integral part of modern entertainment production and biomechanics, as well as being a key research area within Computer Vision. Attempting to accurately find the joint locations of an individual or group of actors within a performance space is a difficult task, but one that is often required in the production of television and film content. Traditionally, this capture is done using marker-based techniques, wherein the performers wear some form of suit or markers that allows cameras to track them. However, this approach is limited by the very markers it uses. As an alternative, markerless techniques have been devised[5][15][21], where performers can use any clothing or costume, freeing them to concentrate on their performance.

Critically, these techniques are constrained to capture volumes; areas where the cameras can record the performance. This is achieved by arranging cameras and other capture equipment in a square or ring around a designated area, and movement outside of this area results in either accuracy reduction or failure to track, due to less cameras being able to view the performer. This effectively creates an "outside-in" system, with a hard limit on the capture volume. We therefore propose a system using a pair of 360° cameras that can track human motion throughout a sequence and then produce a temporally consistent 3D skeleton. Given the natural advantages of omni-directionality, we do not suffer "out of shot" problems in the same sense as a traditional camera setup, and as such, we create a capture system with a pair of cameras at the centre of the scene, and whose capture volume is restricted only by the resolution of the cameras.

We therefore present our work, which aims to make two contributions:

- A robust tracking algorithm that operates in a 360° image space across a pair of cameras, capable of tracking an individual from a multi-person scene
- A 3D skeletal position solver that takes joint estimates from two 360° cameras to produce a temporally consistent body pose in 3D space

2. Related Works

Our proposed system can be split into two stages, the tracking algorithm and the body pose reconstructor. Human tracking can be defined as following a specific individual throughout a scene recorded from one or more cameras, and providing a location of that person either in 2D space (within the frame of a specific camera), or in 3D space (relative to the cameras). More specifically, a tracking algorithm needs to identify potential human shapes within the scene, then track by association of human observations over time. A robust tracking algorithm should be capable of re-identification[8], where tracking of the individual should continue even when the subject cannot be seen, or moves into a different camera view.

Tracking algorithms thus far have concentrated on perspective images, providing limited fields of view from which the subject can easily leave. This contrasts with 360° cameras, where the subject can only leave the camera view through occlusions. However, the large field of view comes at the cost of much reduced resolution, as well as projection issues, which require any tracker to be flexible enough to cope with distortions introduced by projection (such as those that can occur when an individual goes around the rear of the camera).

3D Motion capture, meanwhile, can be defined as taking the movements of a specific individual within a scene and accurately reconstructing the pose at each frame. More specifically, we need to identify the individual body components (as joints, body segments etc.), calculate their 3D

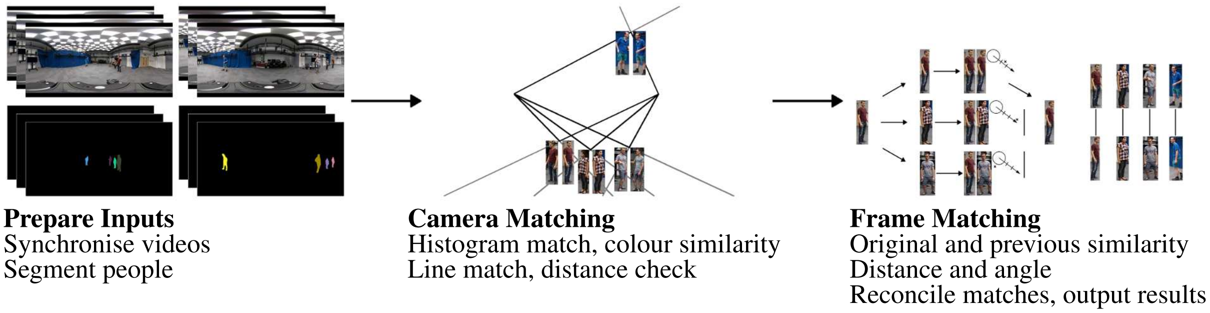
Prepare Inputs
Synchronise videos
Segment people
Camera Matching
Histogram match, colour similarity
Line match, distance check
Frame Matching
Original and previous similarity
Distance and angle
Reconcile matches, output results

Figure 1. System overview of person matching and tracking component

position and use the information to reason about the location of any unidentified points, thus estimating the full 3D pose in each frame.

As with tracking algorithms, motion capture systems have used perspective cameras to define a capture volume. This volume has the advantage of being (generally) well defined, however it suffers from the problem that it needs to be large enough to capture the whole scene. 360° cameras, meanwhile, allow the capture volume to extend out from them, being limited not by the location and fields of view of the cameras, but rather by the resolution of the cameras.

2.1. Human Tracking in Video

Until fairly recently, tracking has been accomplished through hand crafted features. Kahn *et al.* [14] uses a series of cameras with overlapping fields of view to track individuals. This is achieved by determining the field of view limits of each camera then discovering their relative 3D positions. This allows people to be tracked between cameras without relying on colour models or calibration information, although this requires an individual to 'prime' the system before use. Zhou and Hoang[35] perform a background subtraction to identify individuals in the scene, with an attempt to remove shadows cast from the remaining image. They then describe each person in terms colour, direction, velocity and relative size to match them temporally. This system runs in near realtime, although it is limited to a single camera.

More recently, Danelljan *et al.* [6] utilise a joint translation and scale tracker. At a given frame, translation and scale changes are jointly calculated based upon the previous position and scale. The respective translation and scale models are then updated and used in the next frame, giving rapid tracking capabilities on any dynamic component. Taking a different route, Zhang *et al.* [34] make use of an online SVM that is updated after each frame, as well as a series of entropy minimisation systems, each with a different learning rate so as to prevent the system from reinforcing poor results.

As with other fields, deep learning has had a profound impact on tracking algorithms. Nam and Han[22] being one of the first to exploit this with a Multi-Domain Network (MDNet). They use a relatively small 7-layer network, noting that larger networks tended to give lower performance. Insafutdinov *et al.* [13] also propose a deep learning approach, but combined with traditional techniques, creating a combined top-down/bottom-up solution. Joint proposals are made throughout the scene in a bottom up manner, but these are then combined with edge and feasible skeleton proposals, as well as temporal consistency. The work of Fernando *et al.* [9] moved to create a tracker for use in real-time applications. Each frame is processed by a lightweight Generative Adversarial Network, using an object pool to create both a short and long term memory which can then be utilised for trajectory tracking and prediction.

While all of these tracking algorithms are effective, they all suffer from the problem that they are designed to work on perspective images, and as such were not built to deal with 360° images that introduce both distortion and image wrapping.

2.2. 3D Human Motion Capture

Early attempts to capture human motion without the use of markers was made using multi view stereo methods, utilizing numerous cameras in a calibrated fashion[20][24]. Carranza *et al.* [4] takes silhouettes produced from 7 calibrated cameras and optimises a pre-constructed human body model to them, minimising the energy required to move to the next frame to infer temporal consistency. Starck and Hilton[26] perform a visual hull reconstruction using 16 cameras and a chroma-key background, then refine the visual hull using detected keypoints.

De Aguiar *et al.* [7] take calibrated sequences from 8 cameras and fit a pre-scanned mesh to them using keypoints matched across the body. This allows them to capture a performer in loose clothing, although this requires a laser scan of the performer to be completed in advance. Similarly, Vlasic *et al.* [30] also use a mesh fitting approach, although in this case the mesh was fitted to silhouettes and a template skeleton, and requires manual editing after processing to tidy the mesh.

Liu *et al.* [17] moves away from a mesh approach and uses pre-scanned copies of each individual to create an articulated skeleton. These skeletons can then be fitted into 2D segmentations from each image, which contain 2 or more people in close proximity. Thus, the segmentation and skeleton fit are jointly optimised and then used to import the pre-scanned models. Trumble *et al.* [27] applies a deep learning approach, taking multiview stereo data and constructing a probabilistic visual hull to which joint positions are fitted.

Marcard *et al.* [29] expands from traditional multi-view stereo and combined the visual information with Inertial Measurement Units (IMUs), allowing occluded joints to be tracked across all input frames. Trumble *et al.* [28] applies both video and IMU data to a deep learning approach, using Long Short Term Memory to reduce temporal noise, and Malleson *et al.* [19] takes input video and IMU data to create a robust and temporally consistent skeletal fitting system.

Combining depth information with video input has also been used to good effect. Shotton *et al.* [25] takes large quantities of synthetic RGB-D data and uses it to train a series of Random Decision Forests, which in turn are used to provide per-pixel classification from an RGB-D image. Wei *et al.* [31] similarly use a single RGB-D camera to estimate pose by splitting a detected body into 15 rigid segments, with movable joints connecting them. Each pixel is given a likelihood of belonging to a specific section, then a skeleton is fitted to these segments.

Ichim and Tombari[12] also take a single RGB-D camera and attempt to fit a series of body-blend shapes to the frame, combined with temporal consistency from previous frames to explicitly track body parts. More recently, Xu *et al.* [32] developed MonoPerfCap, a system capable of creating a full body model from a single handheld RGB camera, taking an initial mesh of the individual in a t-pose and then applying that mesh to any new footage.

All of these contributions, however, have limitations that cannot readily be overcome when using 360° cameras. Specifically, they require either highly accurate calibration information, and/or depth information. Additionally, the representation of 360° images introduces distortions that these systems cannot account for.

2.3. 360° Imagery

While the above techniques all work well within the perspective images for which they were designed, they generally fail when presented with 360° imagery, either due to distortions introduced or by image discontinuity (such as on the edges of equirectangular or cubic representations). There has been relatively limited work on tracking people in 360° images. More specifically, during our review, no tracking algorithms could be found that specifically operated on 360° images. As such, we will review a range of wide angle and 360° works.

One of the earlier attempts at fisheye reconstruction was Li[16], taking two 190° fisheye cameras to create a disparity map based reconstruction, allowing for basic head movement motion. However, this method requires highly accurate camera calibration to be established, something that can be difficult for full 360° cameras. Chuiwen *et al.* [18], created a full 360° reconstruction using a pair of 360° cameras across a short baseline. Images are projected into cube maps, a small set of matches are manually made, which were then used to guide future matches. This process produces a reconstruction, but suffers from needing both manual correspondences to be established, and for a ground truth calibration to be provided. Fowler *et al.* [10] presented work aimed at providing a human centric affordance map from a single 360° camera. An individual is identified in each frame, with depth estimated from neck length. Then, the individuals activity is broadly identified (walking, sitting, standing *etc.*) and their action, along with position and depth, and used to produce an affordance map, as well as a 3D scene reconstruction of the respective surfaces. It is limited, however, to 2D per frame pose estimation of a single individual and does not reconstruct 3D pose nor handle multiple individuals.

Rhodin *et al.* [23] mount two floor-facing fisheye cameras above the performer, estimating 3D pose using a bespoke ConvNet to estimate local pose, and Structure from Motion algorithms to place that local pose in a global scene. Xu *et al.* [33] expand upon this, using only a single camera mounted on the peak of a baseball cap, and a CNN architecture to estimate pose, albeit without global scene placement. While both approaches free the performer from a studio, both suffer from the performer being constrained with sensitive head-mounted equipment.

3. Methodology

Our proposed method is divided into two components. Initially, two temporally aligned 360° video sequences of a scene with multiple people are taken and individuals are tracked across the 2D image sequence. Once we have the these tracks, joint locations are estimated using OpenPose[3] and are used to create a 3D skeleton relative to the two camera positions.

Initially, we must define the notation for working with 360° images. We describe equirectangular images in terms of $-180° \leq \theta < 180°$ and $-90° \leq \phi \leq 90°$, with the left edge of the image being $\theta = -180°$ and increasing as we move to the right, and the top of the image being $\phi = 90°$, decreasing as we move down the image.

As such, for a given pixel x, y on an equirectangular image of height h and width w, we can find the θ, ϕ angles using eq.1 and eq.2 respectively.

$$\theta(x) = \frac{x}{w/360} - 180 \tag{1}$$

$$\phi(y) = -\left(\frac{y}{h/180} - 90\right) \tag{2}$$

3.1. Person Matching

Our initial problem is one of data matching across multiple views. Given a pair of synchronised video frames f_n^α, f_n^β, where α, β are cameras that are horizontally disjoint, and whose extrinsics are known, and n is the frame number, we want to match the same person p (from the set of all people, P) across both frames. We segment each frame using Mask RCNN[11], producing two sets of segments, S_n^α, S_n^β for f_n^α, f_n^β respectively. For each segment s_n at frame n, we assign an angle θ_{s_n} relative to the camera by taking the x coordinate of the centroid of the segment.

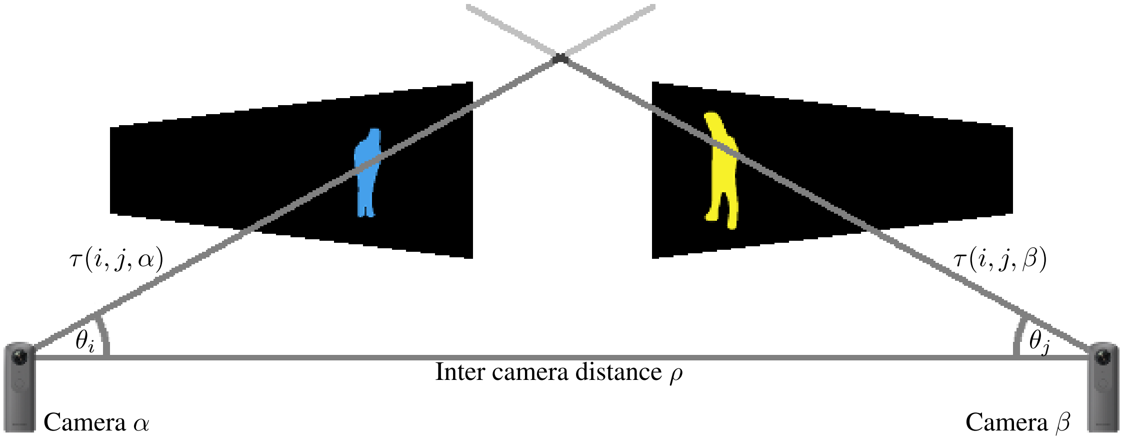

Figure 2. Example of triangulation from cameras α, β of segments i, j

We then match $i_n \in S_n^\alpha$, $j_n \in S_n^\beta$, (where i_n, j_n are individual segments normalised to a constant height and proportional width) using the cost function in eq.3

$$C(i_n, j_n) = col(i_n, j_n) + scale(i_n, j_n) \tag{3}$$

where $col(i_n, j_n)$ is the colour similarity between two segments, i_n, j_n and $scale(i_n, j_n)$ is the distance of the two segments relative to the distance between the cameras.

The colour similarity metric is defined as

$$col(i,j) = col_{glo}(i,j) + col_{reg}(i,j) + col_{loc}(i,j)$$

and measures the similarity of the segments i, j at three different scales, in order to account for differing resolutions of the segment between the two cameras. Generally, these measures work by exploiting the observation that, excluding the face, human subjects are generally similar in appearance irrespective of viewing angle about the vertical axis.

For the global similarity term $col_{glo}(i,j)$, we construct a histogram H^s of $B = 20$ bins from segment s by taking each pixel in hue-saturation-value colour space, and placing the pixel in a bin according to it's hue value. We then compare the two histograms using

$$col_{glo}(i,j) = \Phi_{glo} \cdot \left(\sum_{k=0}^{B} |\eta_k^i - \eta_k^j| + \sum_{k=0}^{B} |\delta_k^i - \delta_k^j| \right)$$

where k is an index, η_l^s is the hue component of bin l of the histogram H^s for segment s, and δ_l^s is the saturation component of bin l of the histogram H^s for segment s. Φ_{glo} is a pre-defined weight applied to the measure, empirically we found $\Phi_{glo} = 2$ to work well.

The region similarity metric first splits the segment s into $b = 4$ equal sized horizontal bands. We then take the median hue μ_ϵ^s and the median saturation ν_ϵ^s for each band, where ϵ is the band number. From this, we calculate $col_{reg}(i,j)$ as

$$col_{reg}(i,j) = \Phi_{reg} \cdot \left(\sum_{k=0}^{b} \sigma(\mu_k^i, \mu_k^j) + ||\nu_k^i, \nu_k^j|| \right)$$

where k is an index, where $||v_1, v_2||$ is the l_2 norm between v_1 and v_2, and $\sigma(\theta_1, \theta_2)$ is the shortest circular distance between two angles as defined in eq.4

$$\sigma_1 = \theta_1 - \theta_2$$
$$\sigma_2 = \begin{cases} \sigma_1 + 360 : \text{for } \sigma_1 \leq 0 \\ \sigma_1 - 360 : \text{for } \sigma_1 > 0 \end{cases}$$
$$\sigma(\theta_1, \theta_2) = |\min(\sigma_1, \sigma_2)| \tag{4}$$

Φ_{reg} is a pre-defined weight applied to the measure, empirically we found $\Phi_{reg} = 1$ to work well.

Our detail orientated metric, $col_{loc}(i,j)$ is defined similarly to col_{reg}. However, rather than compare broad regions, we compare individual rows of the segment. Given a normalised segment height of κ, $col_{loc}(i,j)$ is the proportion of rows that are similar in both hue and similarity

$$col_{loc}(i,j) = \Phi_{loc} \cdot \left(\frac{\sum_{k=0}^{\kappa} l(k)}{\kappa} \right)$$

where Φ_{loc} is a pre-defined weight applied to the measure, empirically we found $\Phi_{loc} = 3$ to work well, and

$$l(k) = \sigma(\mu_k^i, \mu_k^j) < \lambda \wedge ||\nu_k^i, \nu_k^j|| < \zeta$$

with $\lambda = 10$ and $\zeta = 50$ being threshold values and providing good discriminative capabilities.

The second component of $C(i,j)$ is $scale(i,j)$, and is defined as

$$T = max(\tau(i,j,\alpha), \tau(i,j,\beta))$$

$$scale(i,j) = \Phi_{scale} \cdot \begin{cases} T/\rho : \text{for } T < \rho \\ \rho/T : \text{for } T \geq \rho \end{cases} \tag{5}$$

where Φ_{scale} is a pre-determined weight applied to the measure, empirically we found $\Phi_{scale} = 1$ to work well, ρ is the distance between the two cameras α and β, and $\tau(i,j,c))$ is the distance to the triangulation point of θ_i, θ_j from camera c (see figure 2)

3.2. Person Tracking

Once all of the segments in f_n^α, f_n^β have been matched, we then track an individual pairing by comparing properties between frames. We achieve this by using the cost function in eq.6

$$F(s_n, s_{n-1}) = app(s_n, s_{n-1}) + pos(s_n, s_{n-1}) \quad (6)$$

where s_n is a segment at frame n, $app(s_n, s_{n-1})$ compares the appearance of the two segments, and $pos(s_n, s_{n-1})$ compares the position against the expected position.

The appearance metric uses the region matching metric from section 3.1, and is defined as

$$app(s_n, s_{n-1}) = \Phi_P \cdot col_{reg}(s_n, s_{n-1}) + \Phi_O \cdot col_{reg}(s_n, s_0)$$

where Φ_P, Φ_O are pre-defined weights, for which we found 2 worked well for both. This metric therefore balances the relative appearance of the segments throughout the frame, as well as ensuring the appearance matches that of the original segment, so that mis-identification can be minimised.

The position metric is defined as

$$pos(s_n, s_{n-1}) = \Phi_D \cdot D(s_n, s_{n-1}) + \Phi_A \cdot A(s_n, s_{n-1})$$

where

$$D(s_n, s_{n-1}) = \left|\iota_{s_n} - \left(\iota_{s_{n-1}} + \left(\iota_{s_{n-2}} - \iota_{s_{n-1}}\right)\right)\right|,$$

$$A(s_n, s_{n-1}) = \left|\theta_{s_n} - \left(\theta_{s_{n-1}} + \sigma\left(\theta_{s_{n-2}}, \theta_{s_{n-1}}\right)\right)\right|$$

ι_s is the height (in pixels) of a segment, and Φ_D, Φ_A are pre-defined weights applied to the measure, we found 1 and 2 respectively to work well.

An additional check is made that if either $D(s_n, s_{n-1})$ or $A(s_n, s_{n-1})$ is 0, or if $F(s_n, s_{n-1}) < 0.6$, no match will be made.

By using the above matching on both camera images f_{n-1}^x, f_n^x, we produce a matching set Π_n^x for camera x between f_{n-1} and f_n, which we augment with a pairing set between the cameras on frame f, achieved exactly as the initial inter pairing (section 3.1). In the ideal case, our inter-frame matching set and the inter-camera matching set agree, however in the event they do not, the inter-camera matching set with the lowest cost is selected, and the other matching set is rejected. With our correspondence determined, we then perform a triangulation between the θ_s of each segment s in each camera to confirm the match has validity in physical space.

3.3. 3D Skeletal Pose Estimation

Once an individual has been tracked in each frame, we can estimate the 3D skeletal pose. For each frame, we isolate each individual using the segment produced from section 3.2 and estimate the joint locations using OpenPose[3]. This gives us a set of joints $\omega_o^c \in \psi^c$, where c is the camera providing the image, and o is the joint number.

Dataset	MDNet[22]	Ours
Occluded Crate	89.2%	**98.0%**
Double Square 8	55.6%	**95.4%**
Random Walk	35.0%	**96.6%**
Seminar Room	36.8%	**71.0%**
Outdoor 2 people	74.0%	**85.0%**
Outdoor 4 people	56.9%	**58.1%**
Outdoor 8 people	62.8%	**64.9%**

Table 1. Performance of the proposed algorithm

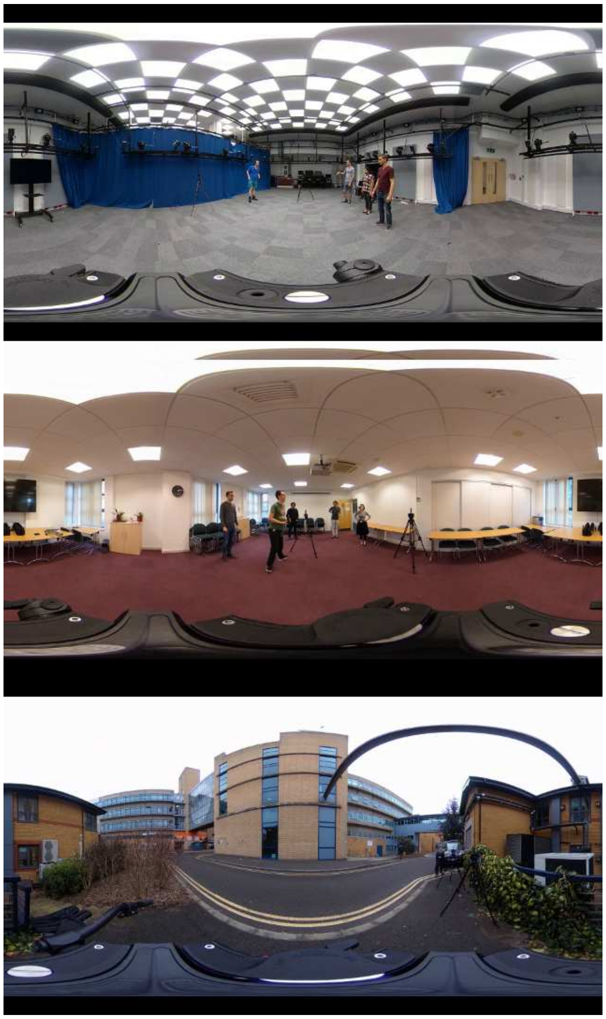

Figure 3. Examples from datasets. Top: Random Walk. Middle: Seminar Room. Bottom: Outdoor 2 people

From these joints, we can then perform a naïve triangulation in all cases where ω_o^α and ω_o^β both exist, producing for each frame f a 3D joint position $\Omega_o \in \Psi$. By taking the distance between specific pairs of o we can estimate bone lengths l_o. After performing this length estimate for each frame, we sort the lengths and remove the highest and lowest 10%, before taking the mean of the remaining lengths. This is done in order to remove outlier bone lengths, such as those produced when the joints are found directly between the cameras. This gives us a reasonable estimate for each bone length specific to the individual being tracked, rather

Dataset	MDNet[22]	Ours
Double Square 8	6	**2**
Random Walk	21	**0**
Seminar Room	32	**6**
Outdoor 2 people	0	**0**
Outdoor 4 people	**0**	1
Outdoor 8 people	**2**	4

Table 2. Number of track "jumps"

than using generic bone lengths or prior body models.

Once we have our joints and bone lengths, we perform a gradient descent optimisation using Ceres[2] in order to find a skeleton pose that best fits the set of joint estimates ω^{α} and ω^{β} while having fixed bone lengths l. We provide the optimiser with a root point in 3D space (from section 3.2) which is initally assigned to the neck point. From this point, we use a series of axis/angle rotations (as per Malleson *et al.* [19]) to describe the joint rotations, and combine this with the respective fixed bone length l_o to create a skeleton scaled to the tracked individual, S_o, where n is the frame number relating to the skeleton estimate.

In order to provide an element of temporal consistency, and to handle error cases (where joints are either not identified, or mis-identified), we use the skeleton from the frame f_{n-1} as the base skeleton for frame f_n. We also project the skeletal estimate back onto the input camera positions, producing the projected joints r^c_o, where c is the camera and o is the joint number.

For each frame, we minimise the cost eq.7

$$\lambda(n) = \lambda_{cv}(n)^s + \lambda_{mv}(n)^s + \lambda_{ac}(n)^s \tag{7}$$

where $\lambda_{cv}(n) = \sum_{o=0}^{O} ||\omega^{\alpha}_o - r^{\alpha}_o|| + \sum_{o=0}^{O} ||\omega^{\beta}_o - r^{\beta}_o||$ is the sum of the l_2 norms between the visible joints ω^c_o and the projected points r^c_o, $\lambda_{mv}(n) = (\sum_{o=0}^{O} ||\omega^{\alpha}_o - r'^{\alpha}_o - \psi^{\alpha}_o||) + (\sum_{o=0}^{O} ||\omega^{\beta}_o - r'^{\beta}_o - \psi^{\beta}_n||)$ is the sum of the l_2 norms between the joints ω^c_o and the projected points from the previous frame r'^c_o, minus the average movement of the joints ψ^c_o over the sequence, and $\lambda_{ac}(n) = \sum_{o=0}^{O} |\sigma(r''^{\alpha}_o, r'^{\alpha}_o) - \sigma(r'^{\alpha}_o, r^{\alpha}_o)| + |\sigma(r''^{\beta}_o, r'^{\beta}_o) - \sigma(r'^{\beta}_o, r^{\beta}_o)|$ is the sum of the acceleration difference between r''^c_o, r'^c_o (the projected joints from f_{n-2} and f_{n-1}) and r'^c_o, r^c_o (where angular difference is calculated using equation 4).

O is the total number of joints (in our case, 18), s is used to scale error to the scene size, empirically we found $s = 2$ to work well for indoor scenes, and $s = 0.5$ for outdoor scenes, with this weight being sensitive to change.

4. Evaluation

We evaluate our algorithms separately, so as to better identify any weaknesses of the individual algorithms. Each algorithm was tested on a series of datasets comprising a range of scenarios, three indoor (studio) scenes, one indoor (non-studio) scene and three outdoor scenes. Each dataset was captured using a pair of Ricoh Theta V[1] cameras, plus a third Ricoh Theta S camera for evaluation purposes only, all placed at a consistent height. A manually annotated ground truth was produced for each dataset.

Of the three studio scenes (figure 3, top), Occluded Crate is a 1 person scene comprising one person walking behind a stack of crates, Double Square 8 is a 3 person scene with 2 people walking and one person in the background, and Random Walk is a 4 person scene with all 4 people walking randomly about the scene. The baseline between cameras is 3.6 metres.

Seminar room (figure 3, middle) is the non-studio indoor scene, comprising 5 people walking randomly about the scene. It contains several difficult occlusions, including double occlusions (where neither camera can observe a given individual). The baseline between cameras is 2.4 metres.

The three outdoor scenes (figure 3, bottom) comprising differing numbers of individuals (2, 4, 6) moving parallel to the cameras. In each case, all individuals are in the front 180° of the camera, due to space restrictions. The baseline between cameras is 3.5 metres.

4.1. Tracking

To assess our algorithm, we compare it's accuracy against both a manually annotated ground truth, as well as against state of the art human tracking algorithms designed for perspective images[22] (since no tracking algorithms were found for 360° videos). Positive frames were defined as those where the method correctly determined the person to track, or that the person did not appear in the frame (due to occlusion), and an overall accuracy was given for a scene from the total number of positive frames as a proportion of the total number of frames.

As shown in table 1, our tracking algorithm gives good results in a studio environment, with an accuracy consistently about 95%, even when subjects are occluded or in scenes containing multiple subjects. Accuracy takes a small drop for our highly crowded Seminar Room dataset. For outdoor scenes, accuracy drops due to the poor resolution of the individuals, combined with similar colours for coats worn. Conversely, MDNet[22] appears to have very poor performance indoors, rising in the outdoor scenes. This is due to MDNet being unable to track the full 360° range, stopping at the image edge. The outdoor scenes, given their 180° nature, confirm this, as accuracy increases on these assessments.

Additionally, we also tracked the number of times the tracker "jumped" from one individual to another. We formally defined a "jump" as any situation instance where the tracker moves onto a different person for more than 15 frames, excluding occlusions. MDNet demonstrated a propensity to "jump" during occlusions (table 2, as well as at the frame extremities (due to nature of the equirectangular representation), and generally didn't return to it's original target unless it moved close to it's current target. Conversely, our tracker remained much more stable even as the

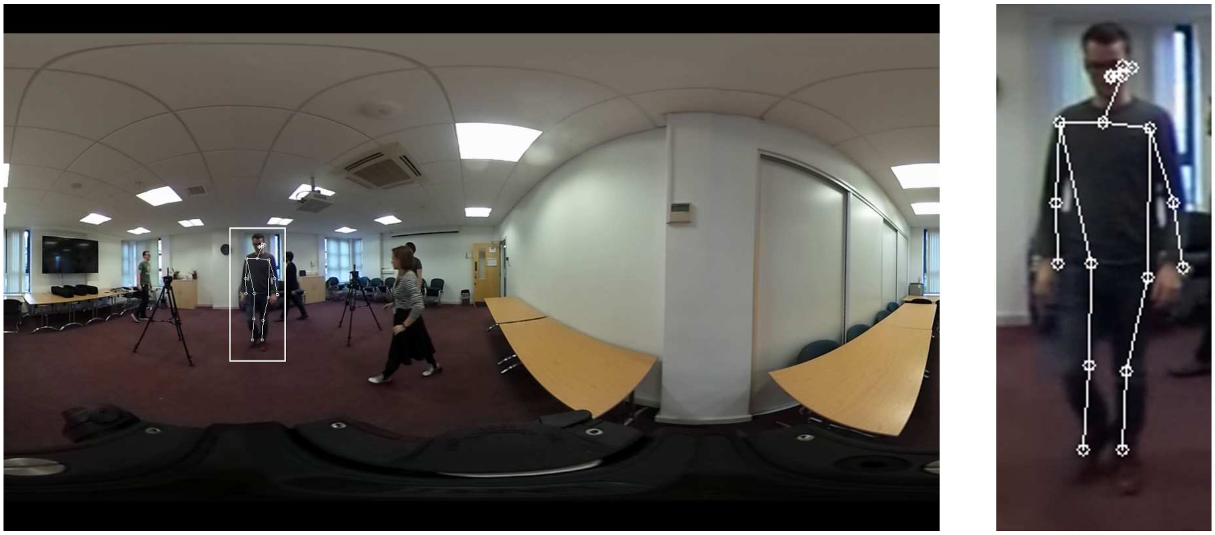

Figure 4. Example of a 3D pose estimate as viewed from the test camera, from Seminar Room

Dataset	Triangulation	Ours
Occluded Crate	**37.03**	55.16
Double Square 8	410.96	**82.36**
Random Walk	228.29	**160.49**
Seminar Room	645.16	**205.06**

Table 3. Average error (pixels) of technique against ground truth

subject became occluded or moved around the "rear" of the camera (*i.e*. moving off the left/right edge of the image and onto the opposite edge).

4.2. Body Position

In order to assess our 3D skeletal estimate, we make use of the ground truth camera, synchronised with the initial two source cameras. As with the source cameras, we estimate the joint positions using Openpose, then project the skeletal estimate onto the ground truth camera. Error is then assessed as the l_2 norm between the projected estimate, and the Openpose joints, normalised throughout the scene to prevent error being linked to distance from the camera. This suffers from the problem that Openpose itself is an estimate, however each ground truth frame is checked for mis-identification before assessment. The results are compared to a naïve triangulation, since no pose estimation systems could be found using only a pair of cameras without additional information (depth, IMU's *etc*.). Our results can be found in table 3, and an example estimate can be seen in figure 4.

5. Conclusions

We have presented a combined tracking and 3D pose estimation system, operating in 360° space and capable of producing realistic 3D poses in both controlled and uncontrolled conditions. Our colour model matching operates well in conjunction with positional matching in order to avoid "jumping" at occlusions, while keeping a colour model from the initial frame stops the tracker reinforcing itself when it does begin tracking the wrong individual. Our pose estimation also works well, producing skeleton estimates that, while not state of the art, are produced from just two cameras.

Our work is not without limitation, however. Our use of colour modelling makes the system less suitable for either crowded scenes or events with large amounts of similarly coloured individuals (such as sporting events). Our 3D pose estimation also suffers from a lack of input, and is highly vulnerable to either mis-detection or no data during occlusions. Our future work should therefore concentrate on these weaknesses, either through the addition of cameras or by improving the colour model.

6. Acknowledgments

This work is supported by both the EPSRC (grant number EP/N 509383/1) and BBC Research and Development. The authors would like to thank the reviewers for their constructive comments.

References

[1] Ricoh Theta V. https://theta360.com/uk/about/theta/v.html, 2019. Accessed: 2019-02-22. 6

[2] S. Agarwal, K. Mierle, and Others. Ceres solver. `http://ceres-solver.org`. 6

[3] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh. OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. In *arXiv preprint arXiv:1812.08008*, 2018. 3, 5

[4] J. Carranza, C. Theobalt, M. A. Magnor, and H.-P. Seidel. Free-viewpoint video of human actors. In *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, pages 569–577, New York, NY, USA, 2003. ACM. 2

[5] S. L. Colyer, M. Evans, D. P. Cosker, and A. I. T. Salo. A review of the evolution of vision-based motion analysis and

the integration of advanced computer vision methods towards developing a markerless system. *Sports Medicine - Open*, 4(1):24, Jun 2018. 1

[6] M. Danelljan, G. Häger, F. Shahbaz Khan, and M. Felsberg. Accurate scale estimation for robust visual tracking. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 2014. 2

[7] E. de Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, and S. Thrun. Performance capture from sparse multi-view video. *ACM Trans. Graph.*, 27(3):98:1–98:10, Aug. 2008. 2

[8] M. Farenzena, L. Bazzani, A. Perina, V. Murino, and M. Cristani. Person re-identification by symmetry-driven accumulation of local features. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2360–2367, June 2010. 1

[9] T. Fernando, S. Denman, S. Sridharan, and C. Fookes. Tracking by prediction: A deep generative model for mutli-person localisation and tracking. *CoRR*, abs/1803.03347, 2018. 2

[10] S. Fowler, H. Kim, and A. Hilton. Human-centric scene understanding from single view 360 video. In *International Conference on 3DVision (3DV)*, 2018. 3

[11] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask r-cnn. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 3

[12] A. E. Ichim and F. Tombari. Semantic parametric body shape estimation from noisy depth sequences. *Robotics and Autonomous Systems*, 75:539 – 549, 2016. 3

[13] E. Insafutdinov, M. Andriluka, L. Pishchulin, S. Tang, E. Levinkov, B. Andres, and B. Schiele. Arttrack: Articulated multi-person tracking in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 4327. IEEE, 2017. 2

[14] S. Khan, O. Javed, Z. Rasheed, and M. Shah. Human tracking in multiple cameras. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 1, pages 331–336 vol.1, July 2001. 2

[15] E. Knippenberg, J. Verbrugghe, I. Lamers, S. Palmaers, A. Timmermans, and A. Spooren. Markerless motion capture systems as training device in neurological rehabilitation: a systematic review of their use, application, target population and efficacy. *Journal of NeuroEngineering and Rehabilitation*, 14(1):61, Jun 2017. 1

[16] S. Li. Binocular Spherical Stereo. *IEEE Transactions on Intelligent Transportation Systems*, 9(4):589–600, 2008. 3

[17] Y. Liu, C. Stoll, J. Gall, H. Seidel, and C. Theobalt. Markerless motion capture of interacting characters using multi-view image segmentation. In *CVPR 2011*, pages 1249–1256, June 2011. 2

[18] C. Ma, L. Shi, H. Huang, and M. Yan. 3d reconstruction from full-view fisheye camera. *CoRR*, abs/1506.06273, 2015. 3

[19] C. Malleson, A. Gilbert, M. Trumble, J. Collomosse, A. Hilton, and M. Volino. Real-time full-body motion capture from video and imus. In *2017 International Conference on 3D Vision (3DV)*, pages 449–457, Oct 2017. 3, 6

[20] T. B. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, 81(3):231 – 268, 2001. 2

[21] T. B. Moeslund, A. Hilton, V. Krger, and L. Sigal. *Visual Analysis of Humans: Looking at People*. Springer Publishing Company, Incorporated, 2013. 1

[22] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2, 5, 6

[23] H. Rhodin, C. Richardt, D. Casas, E. Insafutdinov, M. Shafiei, H.-P. Seidel, B. Schiele, and C. Theobalt. Egocap: Egocentric marker-less motion capture with two fisheye cameras. *ACM Trans. Graph.*, 35(6):162:1–162:11, Nov. 2016. 3

[24] N. Sarafianos, B. Boteanu, B. Ionescu, and I. A. Kakadiaris. 3d human pose estimation: A review of the literature and analysis of covariates. *Computer Vision and Image Understanding*, 152:1 – 20, 2016. 2

[25] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *CVPR 2011(CVPR)*, volume 00, pages 1297–1304, 06 2011. 3

[26] J. Starck and A. Hilton. Surface capture for performance-based animation. *IEEE Computer Graphics and Applications*, 27(3):21–31, May 2007. 2

[27] M. Trumble, A. Gilbert, A. Hilton, and J. Collomosse. Deep convolutional networks for marker-less human pose estimation from multiple views. In *Proceedings of the 13th European Conference on Visual Media Production (CVMP 2016)*, CVMP 2016, pages 6:1–6:9, New York, NY, USA, 2016. ACM. 2

[28] M. Trumble, A. Gilbert, C. Malleson, A. Hilton, and J. Collomosse. Total capture: 3d human pose estimation fusing video and inertial sensors. In *Proceedings of 28th British Machine Vision Conference*, pages 1–13, 2017. 3

[29] T. v. Marcard, G. Pons-Moll, and B. Rosenhahn. Human pose estimation from video and imus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(8):1533–1547, Aug 2016. 3

[30] D. Vlasic, I. Baran, W. Matusik, and J. Popović. Articulated mesh animation from multi-view silhouettes. In *ACM SIGGRAPH 2008 Papers*, SIGGRAPH '08, pages 97:1–97:9, New York, NY, USA, 2008. ACM. 2

[31] X. Wei, P. Zhang, and J. Chai. Accurate realtime full-body motion capture using a single depth camera. *ACM Trans. Graph.*, 31(6):188:1–188:12, Nov. 2012. 3

[32] W. Xu, A. Chatterjee, M. Zollhöfer, H. Rhodin, D. Mehta, H.-P. Seidel, and C. Theobalt. Monoperfcap: Human performance capture from monocular video. *ACM Trans. Graph.*, 37(2):27:1–27:15, May 2018. 3

[33] W. Xu, A. Chatterjee, M. Zollhfer, H. Rhodin, P. Fua, H. Seidel, and C. Theobalt. Mo2cap2: Real-time mobile 3d motion capture with a cap-mounted fisheye camera. *IEEE Transactions on Visualization and Computer Graphics*, 25(5):2093–2101, May 2019. 3

[34] J. Zhang, S. Ma, and S. Sclaroff. Meem: Robust tracking via multiple experts using entropy minimization. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 188–203, Cham, 2014. Springer International Publishing. 2

[35] J. Zhou and J. Hoang. Real time robust human detection and tracking system. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, pages 149–149, Sep. 2005. 2