Seeing Invisible Poses: Estimating 3D Body Pose from Egocentric Video

Hao Jiang
Boston College, USA
hjiang@cs.bc.edu

Kristen Grauman
University of Texas at Austin, USA
grauman@cs.utexas.edu

Abstract

Understanding the camera wearer's activity is central to egocentric vision, yet one key facet of that activity is inherently invisible *to the camera—the wearer's body pose. Prior work focuses on estimating the pose of hands and arms when they come into view, but this 1) gives an incomplete view of the full body posture, and 2) prevents any pose estimate at all in many frames, since the hands are only visible in a fraction of daily life activities. We propose to infer the "invisible pose" of a person behind the egocentric camera. Given a single video, our efficient learning-based approach returns the full body 3D joint positions for each frame. Our method exploits cues from the dynamic motion signatures of the surrounding scene—which change predictably as a function of body pose—as well as static scene structures that reveal the viewpoint (e.g., sitting vs. standing). We further introduce a novel energy minimization scheme to infer the pose sequence. It uses soft predictions of the poses per time instant together with a non-parametric model of human pose dynamics over longer windows. Our method outperforms an array of possible alternatives, including typical deep learning approaches for direct pose regression from images.*

1. Introduction

Wearable "egocentric" cameras are steadily gaining traction—thanks not only to smaller devices, but also the increasing promise of vision and learning technology to transform applications. Head- or chest-mounted cameras, initially perceived as the purview of hard-core life loggers, are now valuable tools for many others. Law enforcement agencies across the US are using bodycams in an effort to promote transparency with the public. Psychologists leverage wearable cameras on infants to gain insights into motor and linguistic development [27]. In healthcare, egocentric vision could move daily-living activity monitoring required for motor rehabilitation from the hospital to the home [16, 21].

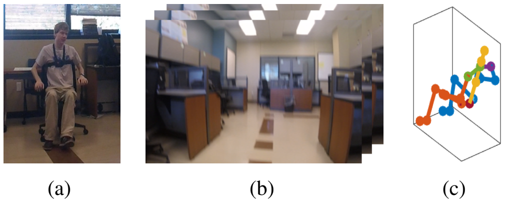

Figure 1. Our goal is to infer the full 3D body pose of a person using the video captured from a single chest-mounted camera. (a): Person with a chest-mounted camera. (b): Egocentric view. (c): Predicted body pose using only video from view (b).

For many applications, the important vision problems center around inferring the camera wearer's behavior, i.e., his activity and interactions with people and objects. As such, the ability to infer the *camera wearer's 3D body pose* is of great interest. However, doing so is challenging because most body parts are invisible to the egocentric camera!

Existing work estimates a person's pose by analyzing the body parts visible in his first-person camera. Naturally, this makes them restricted to the arms and hands [4, 11, 12, 13, 20]. However, from the view of a chest-mounted wide-angle camera, arms and legs are often not visible in daily life activity. For example, in our ground truth videos in which people perform normal activities in public places such as labs and offices, the chance to view arms and legs is less than 10%. To estimate full body pose, one creative approach [1] is to fasten multiple cameras to all the person's joints, then use structure from motion (SfM) to localize the cameras and hence the joints. However, this comes with the disadvantages of requiring 1) obtrusive multi-camera equipment not amenable to everyday casual use and 2) intensive computational requirements (hours to days of processing to infer pose for a minute of video [1]).

We ask the question: *Is it possible to estimate the "invisible" human body pose behind a single egocentric camera?* (See Fig. 1). Despite the fact that we cannot see the person behind the body-mounted camera, the video seen from his point of view provides clues that may well be learnable. In particular, we expect clues from two sources: *dynamic motion signatures* and *static scene structure*. First, there exist motion signatures for pose changes that are resistant to scene changes. For example, the act of standing up has a certain motion pattern as seen by the ego-camera, no matter if he stands up from a chair in a restaurant or a bench at the park. In fact, first-person games use these effects to guide the virtual camera, giving gamers the impression they are moving the same way as the virtual character. Second, static scene structure sets the *context* and offers a prior on likely poses. For example, the pose of typing on a keyboard occurs in similar views showing a monitor or laptop, even though the hands need not be visible. Or, if we see a table in front of us with a specific distance and angle, we can predict whether we are standing or sitting in front of the table.

Of course, not all poses are distinguishable from egocentric video; some will be aliased, meaning different poses can produce the same visual signal. Our intent is to leverage the typical structure linking how the scene changes to how the body is posed. When there is ambiguity, we infer a pose with high probability from the egocentric view.

We introduce a novel approach to predict first-person body pose, given an egocentric video sequence. As training data, our approach takes videos from a wearable camera, where each frame is labeled with ground truth pose parameters. The pose is parameterized by 25 3D joint positions, i.e., a "stick figure" representation, and is obtained with Kinect during training. At test time, we are given a novel RGB egocentric video from a new user, and must infer the sequence of 3D body poses based on the single wearable camera video alone.

Our learning approach capitalizes on the clues described above, while also incorporating longer term pose dynamics. First, classifiers based on dynamic and static cues estimate the probability of each of a (large) set of quantized poses per frame. Then, we jointly infer poses for a longer sequence based on those initial predictions together with a non-parametric model of pose dynamics. The latter is used to identify a least-cost "pose path" through exemplar training video. This step regularizes the initial estimates with priors about how people can move, and is efficiently optimized with dynamic programming. The whole approach is fast—about 0.5 seconds per frame.

We validate our method quantitatively on videos from ten camera wearers performing daily activity poses, as well as qualitatively on challenging videos in unconstrained environments. The experiments show the proposed method gives robust results. It greatly outperforms several alternative methods, including a CNN regression method modeled after the third-person DeepPose [5] approach retrained for our setting.

In summary, our contributions are: (1) We tackle a new problem that estimates the wearer's "invisible" pose from a single egocentric video; (2) We propose a novel global optimization method that leverages both learned dynamic and scene classifiers and the pose coupling over a long time span; and (3) We benchmark several methods, including hand crafted features and CNN learned features, for our task.

2. Related work

We deal with a new problem of predicting invisible human poses from a single egocentric video stream.

Third-person pose Pose estimation from images and video has been studied for decades [7]. Existing work tackles pose estimation from a third-person viewpoint, where the person is entirely visible. In contrast, we consider estimating the body pose of the person *behind* the camera; his body parts are rarely visible, if at all. So, existing pose estimation methods are not applicable to our scenario.

Some third-person pose methods use regression to map from images to pose parameters (e.g., [5, 8, 9, 6]), including the recent DeepPose work using convolutional neural networks [5]. At a glance, a direct regression approach seems like a possible solution for our problem. Even though the body is not visible, we want to learn the connection between what the person sees and how his body is posed. However, a naive application of that idea is inadequate, since 1) even large training sets cannot fully capture the possible variation in environments, poses, and movements, and 2) the relevant egocentric visual signals are inherently temporal. The proposed method learns the connection between pose and dynamic and static cues from snippets of video, and enforces long term constraints between estimated poses. Our experiments show this yields superior results to a DeepPose-like scheme applied to our task.

First-person pose Limited research explores ways to infer the body pose of an egocentric camera wearer [4, 1, 13, 2, 11, 12]. Given interest in understanding handled objects, some methods are dedicated to estimating pixel-wise 2D maps of the camera wearer's hands [13, 11, 12]. Recent work also investigates how depth data from an egocentric RGBD camera can help estimate shoulder, arm, and hand poses in 3D [4], and how specially designed head mounted stereo rigs can be used for markerless mocap [2]. These lines of work assume the body parts are visible in the egocentric view. In contrast, we aim to estimate the full body pose of the person (e.g., 25 joint positions), and we do so even when the body is entirely out of view of the egocentric camera.

In this sense, our goal is more related to the "inside-out" mocap approach of [1]. In that work, 16 or more body-mounted cameras are placed on a person's joints, and then each camera's 3D location is recovered via structure from motion (SfM). There are important differences with our technical approach and motivation. First, rather than 16+ cameras attached at joints worn expressly for the purpose of a mocap session [1], we employ a single chest-mounted camera—the sort typical wearable-computer-users may wear anyway while going about daily activities. Thus, the SfM approach cannot be directly applied to our setting, and our system requirements are more lightweight and flexible. Secondly, our approach is novel. Whereas the mocap method employs a geometric solution to localize the joints, we devise a *learning* solution that discovers the connection between how the egocentered scene changes as a function of body pose. The possible disadvantage of our method relative to [1] is our need for representative training data, though the data is relatively easy to collect, given that it requires no manual annotations (see Sec. 3.1).

Egocentric activity analysis Most recent egocentric vision work studies activity recognition [14, 17, 21, 22, 23, 24, 15] or object recognition [20, 13]. Once again, the focus is largely on visible activity happening in front of the camera—particularly hand-object manipulation activities. However, some work shows that ego-actions (like riding a bus, snowboarding, etc.) are detectable from the scene video [23, 17], and the walking style of the camera wearer can even aid person identification [10] or visual SLAM [18, 19]. We consider whether ego-video can go further to reveal full 3D body pose. While we also use movement information, our method does not infer action classes. For instance, rather than recognize the current action as "walking", our approach will produce the detailed pose across the walking cycle. Thus, our method provides a mid-level representation—explicit pose—which could be further used in high-level activity recognition or other applications.

3. Method

We estimate 3D human poses from a chest-mounted camera. Predicting human poses from egocentric video is a regression problem: from the input video, we estimate the 3D position of each body joint. Next, we show how to compute instantaneous pose estimates using local features and full sequence estimation using our pose path method.

3.1. Pose parameterization and data collection

We use a Kinect V2 sensor to capture the ground truth human poses. Pose is represented as the 3D positions of 25 body joints defined in the MS Kinect SDK. The predicted 3D pose is positioned in a local coordinate system as shown in Fig. 2. The origin is at the center between two hip joints. The first axis is parallel to the ground and points to the wearer's facing direction. The second one is parallel to the ground and the vertical plane passing the shoulder line. The third axis is perpendicular to the ground. The joint coordinates are normalized by five times the shoulder width of the subject. Note, poses are not aligned to the torso. The local coordinate system allows the torso to lean in any direction or rotate along the axis.

We choose chest-mounted (vs. head-mounted) because it provides a stable view unaffected by constant head bobbles. The frame rate of both the Kinect sensor and the ego-camera is 30Hz. The two are synchronized using time stamps. We capture a total of 18 ground truth videos, in which 3 videos are for training and the rest for testing. Ten subjects with different height, body shape, and gender are involved in data collection. They are instructed to perform normal daily activities in public places such as offices, labs, and libraries. The dataset is collected indoors due to the limitation of the Kinect V2 sensor. However, our approach is general, and we demonstrate outdoor tests as well.

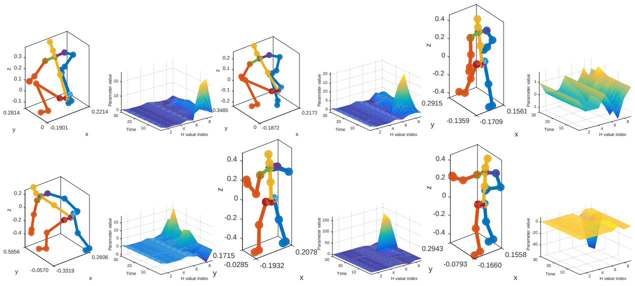

Figure 2. Example poses and the corresponding dynamic features for the surrounding 1-second video segment. Similar poses often have similar dynamic features (see first two examples), and distinct poses have different features. We see that global scene motion gives valuable cues about the coarse body poses of the wearer.

3.2. Instantaneous pose estimation

We first estimate the probability of poses at each time instant. Let function $f(v, p)$ be the probability of video segment v corresponding to pose $p \in P$, where P is the set of all possible poses in the training sequence. Instead of directly computing f, we train a classifier to obtain the function $g(v, c)$ to extract the probability of video segment v matching the pose cluster c. Each pose cluster is represented using the cluster center, which is a vector of joints. The mapping f is approximated as $f(v, p) = g(v, c(p))$, where $c(p)$ is the pose cluster identity of a pose p.

3.2.1 Dynamic clues

Egocentric video shows different dynamic patterns for different movements of the wearer. We extract the sequence of homographies between successive video frames to quantify the video dynamics. Strictly speaking, the homography is scene invariant only when the camera is purely rotating. It is still a body movement representation that is resistant to scene changes when both rotation and translation are involved. This allows us to use very few training data to obtain good classifiers (as opposed to attempting to learn solely appearance-specific cues, which would be overly restrictive to a given training environment).

To compute a homography between frames, we use optical flow to find the point correspondence. A least squares method is used to estimate the homographies using SVD. The elements in each homography are then normalized by the top-left corner element. The stack of normalized homographies over a fixed time interval (one second), is used to represent the camera movement. Fig. 2 illustrates how the

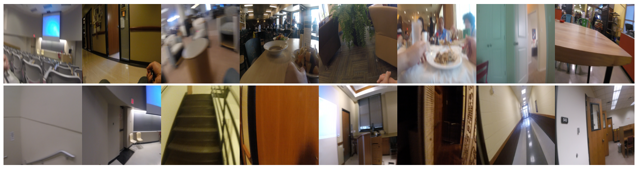

Figure 3. Samples from the training dataset of sitting (Row 1) and standing (Row 2).

proposed feature helps differentiate poses of the wearer.

Using the above feature, we train a random forest to predict the probability of the pose at each instant of the input video belonging to each of the pose clusters. The dynamic feature classifier gives reasonable results. However, the result is ambiguous when there is little motion in the egocentric video. To resolve this issue, we also use static scene structure (context), as defined next.

3.2.2 Static scene structure clues

Apart from dynamics of the scene, the static scene structure also indicates likely human poses. In everyday life, many human poses can be classified as standing-like or sitting-like, e.g., walking is standing-like and kneeling is sitting-like. Indeed, in the dataset in [17], roughly 95% of frames can be classified as standing-like or sitting-like.

We collect a training dataset containing 5,530 standing images and 2,946 sitting images in different indoor environments from subjects of different heights. Fig. 3 shows sample images from the dataset. We train a CNN classifier to categorize each image as representing a sitting- or standing-like pose by fine tuning the last three layers of the fully connected network in AlexNet [26]; the learning rates of other layers are set to zero. The two-class classifier generalizes well. On our ground truth dataset with 71,623 egocentric video frames and poses from the Kinect V2, the sitting-like and standing-like image classification accuracy is 65.09% and 77.97%, respectively. The dataset is composed of 80% images with standing-like poses.

3.2.3 Local cost of pose estimation

Thus far we have provided two ways to quantify the pose probability for each frame, using dynamic and static cues. These instantaneous measurements alone are not sufficient to give the final output of our system, however. As we will explain in Sec. 3.3, errors can be corrected in a global optimization stage where we infer the entire *pose path* over the entire sequence.

In particular, the two classification outputs above serve as unary terms of an energy function for the longer sequence of surrounding frames (1-3 minutes per clip in our dataset). Let $x_{i,n}$ be an indicator variable, which is 1 if at time n the pose i is predicted. Here i is a pose in P. P is the set of all the poses, represented as joint position vectors, in the training sequences. Let $e_{i,n}$ be the cost of predicting pose i at time n. The overall unary cost term is $U = \sum_{n=1..N, i \in P} e_{i,n} x_{i,n}$, where N is the number of frames. We define the cost $e_{i,n} = 1 - g(v_n, c(i)) + d_{i,n}$, where g is the probability of dynamic feature v_n being classified as pose cluster $c(i)$. We use $d_{i,n}$ to penalize the selection of pose i at time n if there is large chance that the the pose class of i and the static scene estimation mismatch. Specifically, we define d as: $d_{i,n} = \delta$ if $h_n > \tau$ and $\hat{g}(i)$ is standing, or $h_n < 1 - \tau$ and $\hat{g}(i)$ is sitting, and otherwise 0. Here h_n is the probability of sitting from the static scene feature at time n. The $\hat{g}$ indicates whether pose i in the training sequence is sitting or standing-like.

Simply optimizing the local cost is not sufficient. Without considering the inter-frame pose constraints the pose predictions can be noisy. Another issue is the resolution. Since the local pose cost is estimated from the probability of quantized poses, it tends to be a staircase function over time. In the following, we show how to solve both of these problems by optimizing poses over a long time span.

3.3. Non-parametric prior on pose dynamics

Next we show how we optimize the final sequence of pose estimates based on the local costs and a non-parametric prior on pose dynamics. First we define the prior, then we introduce an efficient optimization approach.

3.3.1 Pose paths in an implicit motion graph

To infer a likely sequence of poses over time, our method constructs an implicit motion graph that controls the possible transitions between poses in the exemplar training videos. The graph nodes correspond to poses in exemplar videos. The edges indicate possible transitions from one pose to another.

The optimal pose sequence corresponds to the optimal *pose path*. The pose path is composed of a sequence of “steps”, each of which represents a transition from one pose to the next. We enforce that each step can only move from a pose cluster to the same pose cluster or a direct neighbor pose cluster. We define pose clusters as direct neighbors if we can find two poses that are drawn from each of the two pose clusters and are adjacent in time in the exemplar pose sequence. Since the same pose cluster may appear at different times in the exemplar pose sequence, the above rule allows large jumps. To further regularize the pose path, we constrain the step sizes, uniformity of the step sizes, and control the stationary steps on the pose path (see below). Therefore when determining where a step should lead to, we also have to consider previous decisions on the pose path. Thus, the transition costs dynamically change with the traversal history.

This graph is reminiscent of motion-graphs used for motion synthesis in computer graphics [25, 3]. However, whereas motion synthesis aims to generate convincing movements within an annotated mocap database based on a few user-specified anchor poses, our task is to jointly infer the sequence of poses in a novel egocentric video. Another implicit model has also been used in third person people tracking [29]. Unlike traditional motion graphs and implicit models, edge weights in our graph dynamically change to allow the regularizers mentioned above.

We use E to represent the concatenation of all the training pose sequences from the training dataset. The poses in E thus preserve the original temporal order. Selecting a sequence of poses from E is equivalent to find a path on E so that the following energy function is minimized:

$$\min_{\mathcal{X}} \{U(\mathcal{X}) + T(\mathcal{X}) + V(\mathcal{X}) + S(\mathcal{X})\} \qquad (1)$$

s.t. $\mathcal{X}$ represents a sequence of poses drawn from E.

Here $\mathcal{X}$ is the matrix $[x_{i,n}]$, where n is the time index and i is the index of poses in E and recall that $x_{i,n}$ is a binary variable to indicate whether pose i is selected at time n. To represent a path, at each time instant n, we have $\sum_i x_{i,n} = 1$. Here $U(.)$ is the unary term defined in the previous section. $T(.), V(.), S(.)$ are terms that control coupling between poses in the whole sequence. $T(.)$ constrains the step size between successive footprints on the path, $V(.)$ controls the speed of the pose transition, and $S(.)$ restricts stationary steps, all defined next.

3.3.2 The step size term T

If we choose pose $l \in E$ at instant $n-1$, we say we step on point l at time $n-1$. At time n, we may step to $l+k$, where k is the step size from time $n-1$ to n. Since the original exemplar video is continuous, the smaller the k the smoother the pose transition is likely to be. If the step size is 0, we keep the same pose in the time interval. The stationary step can be used to infer a slower movement in the testing video. If the step size is 1, the movement has the same speed in the training and testing video. For $k > 1$, the movement in the testing video is faster than the exemplar sequence. In the energy function, we prefer the step size to be small and at the same time we allow occasional large jumps from one point to the other.

In particular, $T = \sum_{i,j,n} w_{j,i} x_{j,n-1} x_{i,n}$, where $w_{j,i} = 0$ if $i - j \leq 2, i \geq j$ and otherwise $w_{j,i} = \delta$, where δ is a positive constant penalizing backward steps and steps that are greater than two. Apart from the step size constraint, we also constrain that if pose cluster $c(i) \neq c(j)$ and $c(i)$ and $c(j)$ are not consecutive in the training video $w_{j,i} = +\infty$. Here $c(i)$ is the pose cluster of pose i. This prohibits the path from going from one pose to another with too much difference or using a transition of pose clusters not seen in the exemplars. However, it does allow long jumps from one pose cluster to the same pose cluster or one that is a direct neighbor to the cluster. However, such long jumps do have a penalty. So, we prefer that steps on the path move to a directly adjacent frame if possible. We allow the path to go forward or backward.

3.3.3 The speed smoothness of the path V

The above step size term roughly enforces a first order constraint on the path: small steps are taken when possible. However, the path may still have a non-uniform speed of steps in a short time span, which is undesirable because within a time of 1 or 2 seconds human body motion is usually uniform. We thus introduce a second order term to penalize the speed changes:

$$V = \sum_{i,j,n} q(|s_{j,n-1} - (i-j)|) x_{j,n-1} x_{i,n} \ , \qquad (2)$$

where $s_{j,n-1}$ is the speed at time $n-1$, for step j. Here q is a truncated linear function: $q(x) = \mu x$ if $x < \gamma$ and otherwise $q(x) = \mu\gamma$, where γ and μ are constant parameters. This term encourages the path to maintain a constant speed.

3.3.4 The stationary step penalty S in the path

Simply minimizing the first order and second order smoothness of the path is not enough. Recall that the local cost in short time intervals tends to be constant. The steps in the pose path thus tend to be stationary because the first and second order smoothness terms will be zero. The step size penalty helps but is not sufficient. We thus penalize stationary steps:

$$S = \sum_{i,j,n} r(u(j,n-1), i) x_{j,n-1} x_{i,n} \ , \qquad (3)$$

where $r(u(j,n-1),i) = 0$ if $i \neq j$, otherwise $r(u(j,n-1),i) = t(u(j,n-1)+1)$. We therefore count the number of stationary steps and penalize the pose to stay unchanged for a long time. Here, $u(i,n)$ is the number of stationary steps accumulated at time n if the current pose is i; $u(j,n-1)$ is similarly defined. Similar to q, $t(.)$ is a truncated linear function. The stationary step penalty term thus makes the path less likely to stay at one point and helps resolve the temporal resolution loss problem.

3.3.5 Optimizing the pose path

Let $H(i,n)$ be the optimal energy of a pose path if the path ends at a specific pose i at time n. The dimension of H is the number of nodes (pose) in the pose graph × the number of frames. We can rewrite the problem into a recursion:

$$\begin{aligned} H(i,n) =& e_{i,n} + \min_{j \in D_i} \{H(j,n-1) + w_{j,i} + \\ & q(|s(j,n-1) - (i-j)|) + r(u(j,n-1),i)\} \end{aligned} \qquad (4)$$

where $u(i,n) = u(j^*,n-1) + 1$, if $j^* = i$ and otherwise $u(i,n) = 0$, $s(i,n) = i - j^*$, $p(i,n) = j^*$, and $j^* = arg \min_{j \in D_i} \{H(j,n-1) + w_{j,i} + q(|s(j,n-1) - (i -$

$j)|) + r(u(j, n-1), i)\}.D_i$ is the set of poses that can transform to i. $u(i,n), s(i,n), p(i,n)$ are the stationary step number, speed of steps and previous optimal pose selection of the optimal pose path ending at pose i at time n. We initialize $H(i,1) = e_{i,1}$, $u(i,1) = 0, s(i,1) = 0, \forall i \in E$. All the other H are initialized to be $+\infty$, and p to be -1. We can verify that solving the recursion is equivalent to optimizing the pose path energy in Eq. 1. The recursion can be efficiently solved using dynamic programming (DP).

It helps to visualize the optimization in a trellis. The trellis contains M columns and N rows, where M is the number of poses in E and N is the number of input video frames. Each edge corresponds to one possible step in the path. Each node has a cost $e_{i,n}$, where i is the column and n is the row of the trellis. Each edge has a weight $w_{j,i} + q(|s_{j,n-1} - (i-j)|) + r(u(j,n-1), i)$. The DP finds a minimum cost path in the trellis. Note that the edge weights dynamically change in the path finding.

Solving the DP involves updating the state variables H, s, u, p in each node. Since only the nodes inside the same or neighboring cluster are connected by each stage of the trellis, the complexity is much lower than $O(M^2N)$. Moreover, we can use the local pose probability to prune impossible nodes from the trellis. In fact, most of the poses have near zero probability from the random forest classifier. If we only keep nodes that correspond to poses that have probability greater than 0.01, the trellis becomes very sparse and the corresponding DP can be quickly completed (typically contributing 0.01 seconds per frame for our whole system, which takes 0.5 sec/frame).

We stress that our method learns more than 3D camera pose, thanks to the scene structure cues and pose-path prior. It estimates detailed body poses and their transitions at each time instant.

4. Experimentation

We evaluate the performance of the proposed method on both a ground truth dataset and challenging videos in unconstrained environments.

In our ground truth data, the 3D human poses are captured from the Kinect V2 for ten human subjects. The synchronized egocentric video is from a chest-mounted GoPro camera. Below we consider two settings. In the first setting (GT1), training and testing videos are from the same human subject, but taken in disjoint indoor environments such as lab, office, hallway and living room. In the second setting (GT2), the training and testing videos are from different human subjects *and* recorded at different locations. There are in total 71,623 test video frames (about 40 minutes) in the ground truth experiments, consisting of clips ranging from 1-3 minutes each, and 7k-10k training frames. We also test about 15 minutes of video from unconstrained video, which lacks ground truth for evaluation.

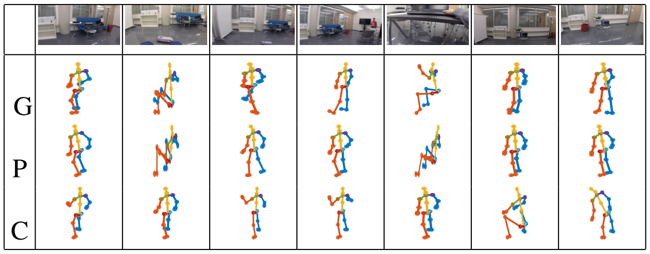

Figure 4. Comparison with the DeepPose [5] method retrained for our task. G: ground truth. P: proposed method. C: `CNN-Regression` baseline.

GroundTruth	Ours	V1	V2	V3	C1	C2	D1	D2
97.5	**76.3**	70.4	8.75	53.8	37.4	27.6	3.2	0.4

Table 1. Percentage of 3D poses that are consistent with human observers' thought. The numbers show the percentages. Ours: Path, V1: Path-Cluster, V2: CNN-Class, V3: CNN-Class-R, C1: KdTree, C2: CNN-Regr, D1: AwaysStanding, D2: AwaysSitting.

	Ours	V1	V2	V3	C1	C2	D1	D2
Head	15.8(8)	16.5(8)	21.6(14)	22.9(14)	18.1(11)	16.2(10)	**15.1**(8)	32.5(9)
Elbow	**14.4**(7)	15.4(7)	18.6(12)	19.4(12)	15.8(10)	14.4(9)	14.5(8)	20.7(8)
Wrist	**19.1**(9)	20.6(10)	26.5(17)	27.1(17)	21.3(13)	22.0(14)	22.9(12)	21.3(8)
Knee	**15.4**(9)	17.2(9)	27.3(17)	26.2(17)	22.0(14)	21.3(13)	21.2(11)	40.0(11)
Ankle	**20.7**(10)	22.9(10)	33.8(21)	33.3(21)	28.4(18)	26.4(17)	26.7(13)	37.9(9)
NAvgAll	**17.2**	19.1	48.1	48.7	32.8	29.7	24.6	31.9
NAvgWA	**19.9**	22.6	60.0	60.2	40.8	38.7	32.4	27.1

Table 2. Average joint error (cm) and standard errors (scaled by 100), when training and testing on same subject but in different environments. See Table 1 for the column labels. The training sequence has 6,950 frames. There are 7 test videos with a total of 25,195 frames. We compute the mean error normalized by the standard error for the nine joints denoted NAvgAll, and for the wrists and ankles denoted NAvgWA.

	Ours	V1	V2	V3	C1	C2	D1	D2
Head	16.6(7)	18.0(7)	19.4(9)	21.3(10)	20.1(9)	15.8(7)	**14.3**(7)	29.1(7)
Elbow	15.3(6)	16.9(6)	19.1(9)	19.5(9)	18.0(8)	15.8(7)	**14.9**(6)	20.9(6)
Wrist	**22.2**(8)	24.2(8)	29.7(14)	29.4(14)	24.9(12)	24.3(11)	23.8(9)	22.9(7)
Knee	**18.9**(7)	24.4(9)	21.6(10)	21.8(10)	31.9(15)	27.6(13)	21.7(8)	45.7(9)
Ankle	**24.9**(9)	29.9(10)	29.2(14)	29.2(14)	38.1(18)	33.3(15)	28.2(10)	43.0(9)
NAvgAll	**19.9**	24.6	35.4	36.4	44.5	34.6	22.4	32.9
NAvgWA	**23.6**	28.4	46.6	46.3	53.3	44.6	28.9	30.7

Table 3. Average joint errors (cm) and standard errors when training and testing on disjoint people and environments. Column labels are defined in Table 1. The training sequence has 10,000 frames from two subjects. There are 8 test videos with a total of 46,428 frames. Table 2 defines NAvgAll and NAvgWA.

	Head	Elbow	Wrist	Knee	Ankle	NAvgAll	NAvgWA
G1	25.0(13)	17.4(9)	21.3(10)	21.1(11)	24.7(11)	26.6	25.7
G2	22.5(9)	19.6(7)	25.7(8)	24.4(8)	29.3(9)	27.0	28.8

Table 4. Two-Cluster baseline (D3) joint errors on the ground truth setting one (G1) and two (G2). See Tables 2 and 3 for definitions of NAvgAll and NAvgWA and units.

Implementation details In the experiment, we use 300 pose clusters. For the unary term U, we set $\delta = 0.1, \tau = 0.99$. For the truncated linear functions q and t, we fix $\gamma = 10, \mu = 0.01$ and $\gamma = 5, \mu = 0.02$, respectively. All parameters were set based on manual inspection of a few examples during method development, then fixed for all experiments. With sufficient labeled data, their values could be set with DP to minimize pose errors.

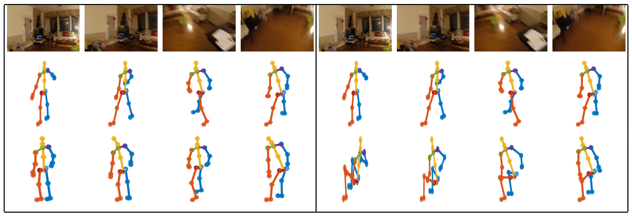

Figure 5. Comparing with the `Kd-tree` baseline. Row 1: Sample frames. Row 2: Ground truth poses. Row 3: Our result in left box and Kd-tree result in right box. Best viewed on pdf.

Baselines No prior work predicts full body pose from a single egocentric video. We therefore devise a series of informative baselines to gauge the impact of our method, including methods inspired by today's best third-person pose estimators:

- **`CNN-Regression`** (C2): an adaption of the DeepPose [5] method to our task. Our problem is still a regression problem, even though the camera wearer is not visible from the egocentric view. We use the same network structure as DeepPose except that our input is a stack of grayscale images in every one-second video clip and output is the 25 body joints defined by the Kinect SDK. We scale each image to 100 × 100. We properly normalize all coordinates for the CNN sigmoid layer.
- **`KdTree`** (C1): simple nearest neighbor approach using Kd-trees. It finds the "closest" video segments in the training data and then takes the corresponding 3D poses as the prediction result. The stacked homography in every 30 frames is used as the feature, and the L_2 norm is the distance metric.
- **`Path-Cluster`** (V1): a variant of the proposed method. Instead of directly optimizing the poses, this method first finds the pose clusters and then refines the pose estimates using dynamic programming. The refinement is similar to the proposed pose path optimization, except that the pose candidates at each instant can only come from the pose clusters estimated in the first stage.
- **`CNN-Class`** (V2, V3): a variant of the proposed method that uses deep-trained features in place of our hand crafted homography features. We train a deep neural network to classify each sequence of 30 frames to one of the 300 pose clusters. We use AlexNet [26] due to its good results in many applications. In the first setting (`CNN-Class`), we rescale each input video frame to 100 × 100 and retrain the network from scratch. In the second setting (`CNN-Class-R`), we fine-tune on the modified AlexNet with depth 30. The fine-tuning is only on the first convolution layer and the last three fully connected layers. We compute the local pose cost as one minus the class probability from the CNN output. The proposed global optimization is then applied to obtain the final result.
- **`AlwaysStanding`**, **`AlwaysSitting`** and hybrid approach (D1, D2, D3): simple guessing methods that exploit the prior that poses are typically somewhere near a standing or sitting pose (hence much stronger than a truly random guess). We compute the standing and sitting poses by the average over training subjects. The hybrid baseline is a variant of path-cluster method that only uses two pose clusters (sitting and standing).

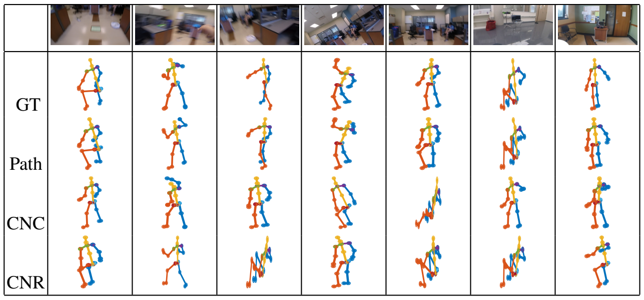

Figure 6. Comparison with methods using deeply learned features. GT: ground truth. Path: proposed method. CNC: `CNN-Class`. CNR: `CNN-Class-R`. Best viewed on pdf.

Comparison to pose baselines: Figs. 4 and 5 show qualitatively that the proposed method indeed gives better results than the DeepPose adaptation (`CNN-Regression`) and nearest neighbors (`KdTree`). Please also see project webpage for video examples.

If we directly use the the estimated pose cluster centers as the predicted poses, the results have lower temporal resolution than the proposed method. Refining the pose selection in each estimated pose cluster is inferior to the proposed approach because the errors in the first stage cannot be undone. The predicted pose sequence is also not as smooth as the proposed method. `Path-Cluster` is essentially an interpolation method that smooths the cluster centers estimated in the first step. Note that a simpler linear interpolation method is not directly usable because it does not always give valid poses.

Fig. 6 shows qualitatively that using deep neural networks to train the dynamic and scene structure features does not give better results. Neither training from scratch nor fine-tuning improves the result. Neural network approaches need a large dataset to capture different variations of the scene and human poses, yet due to the complexity of learning pose from the surrounding scene, CNNs remain inferior to our approach even with 5× more training data. Our method achieves good performance even if training on a relatively small dataset.

Now we present the quantitative comparisons with all baselines. We analyze the errors of the joints with highest variance in everyday activity: head, elbows, wrists, knees, and ankles. We quantify error by the distance between the predicted 3D joints and the ground truth. Recall that the predicted coordinates are already in normalized coordinates according to the shoulder length of the subject. We convert raw errors to centimeters based on a reference shoulder joint distance of 30 cm.

Tables 2 and 3 show the results, for the two settings defined above. Overall the proposed method gives smaller errors than all the competing methods. While `AlwaysStanding` is a reasonable prior for most test frames, our method still makes noticeable gains on it, showing our ability to make fine-grained estimates (e.g., 6 cm better on average for the ankles and knees). `AlwaysSitting` has much larger errors than any method, in line with the distribution of the test data. Table 4 shows that the hybrid approach is also inferior. Finally, among Table 2, 3 and 4, as expected we see that absolute error is lower for all methods with the benefit of observing the same subject during training.

Qualitative perceptual result: We conduct a user study to obtain a human perceptual result on pose inference from egocentric video. The interface shows egocentric video and the 3D poses side by side on a screen. Each 10-second clip is randomly extracted from the ground truth video one and two. Then the output from the Kinect sensor, our proposed method, and all the competing methods are randomly selected. The video and the pose sequences are played at the same time. The users have no knowledge about the source of each pose sequence and need to answer the question about whether the poses are consistent with their thought on the video clip. We invited 30 randomly selected users with different backgrounds to conduct the test. Each user is given 120 video clips in the test.

Table 1 shows how often human observers think the poses and the videos are consistent. The result shows that the ground truth pose from the Kinect sensor is almost always consistent with human observation. The ground truth itself thus has high accuracy and it is a good benchmark to evaluate other methods' results. Our proposed method also has much higher user evaluation score than the competing methods. These are consistent with our ground truth quantitative results.

Comparison to SLAM: We also compare our method against a SLAM (simultaneous location and mapping) approach that estimates the egocentric camera's position and rotation via the video sequence. Note that the trajectory of the camera only gives one point on the chest of a wearer. The question is whether the single camera 3D poses recovered from a state of art SLAM method [28] can be used to infer the full body 3D poses effectively. To estimate the 3D poses, we first cluster trajectory segments in every one-second interval of the training samples using K-means, with the starting point of these 3D trajectories being normalized to the origin. Then a method similar to `CNN-Class` is used to infer the poses. The SLAM based method gives average normalized error of 38.14 in GT1 and 35.77 in GT2 for the 9 joints (head, elbows, wrists, knees and ankles), while our approach's errors are 17.2 and 19.9 respectively. The results of the pure SLAM and the camera motion approach (KdTree) show that *simply estimating the egocentric camera's trajectory or movement is not sufficient to infer accurate 3D human poses*. It is key that our approach uses both the information from camera motion and the scene context to achieve accurate results.

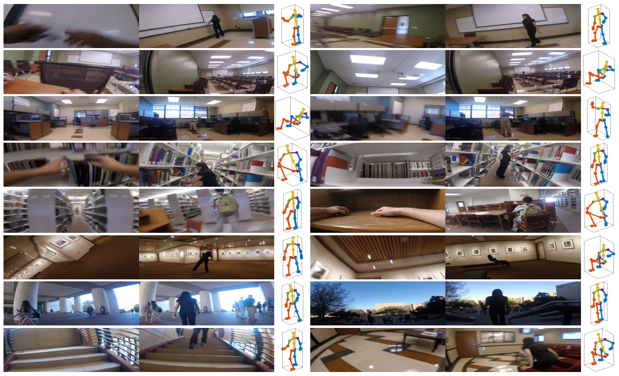

Figure 7. Experiments on data without ground truth. There are three subjects (S1, S2, and S3). Row 1: Classroom (S1). Row 2: Classroom (S2). Row 3: Lab (S3). Row 4: Library (S1). Row 5: Library (S2). Row 6: Art gallery (S1). Row 7: Outdoor (S1). Row 8: Hallway (S1). Each result contains three columns: egocentric view, side view (unseen by our method), and pose prediction.

Our method could fail just like a human observer when estimating human pose by only looking at the egocentric video. Failures are mostly due to the ambiguity of the input. Arm poses are not always predictable, if they do not affect the motion or the viewing angle of the egocentric camera.

Application to unconstrained video: Finally, we test our method on 8 video sequences with no ground truth, captured in varying environments and with 3 subjects. The training dataset is the same as above. Fig. 7 shows sample results. For each example, we display the frame from the egocentric camera as well as one from a side camera viewing the subject. Note that the side view is for display only, and never used by our method. The 3D pose is estimated using only the egocentric video. Our method works well on this data, including an outdoor test sequence despite all training taking place indoors. Please see videos on our project webpage.

5. Conclusion

We tackle a new problem in computer vision: predicting human poses from egocentric video. The proposed global optimization method is able to give accurate pose predictions in both same-person and cross-person tests. Our experiments show our method gives results superior to a number of alternative approaches. We believe our method will be useful for many different applications including egocentric video logging, summarization, and information retrieval, and it could facilitate action understanding.

Aknowledgements: This research is supported in part by U.S. NSF 1018641 (HJ) and ONR PECASE N00014-15-1-2291 (KG).

References

[1] T. Shiratori, H.S. Park, L. Sigal, Y. Sheikh, J.K. Hodgins. Motion Capture from Body-Mounted Cameras. ACM Transactions on Graphics, Vol. 30, No. 4, July 2011.

[2] H. Rhodin, C. Richardt, D. Casas, E, Insafutdinov, M. Shafiei, H.-P. Seidel, B. Schiele, C. Theobalt. EgoCap: Egocentric Marker-less Motion Capture with Two Fisheye Cameras. Siggraph Asia, 2016.

[3] O. Arikan, D.A. Forsyth., J.F. OBrien. Motion Synthesis from Annotations. Siggraph 2003.

[4] G. Rogez., J.J. Supancic, D. Ramanan. First-person Pose Recognition using Egocentric Workspaces. CVPR 2015.

[5] A. Toshev, C. Szegedy. DeepPose: Human Pose Estimation via Deep Neural Networks. CVPR 2014.

[6] C. Sminchisescu, A. Kanaujia, D.N. Metaxas. BME : Discriminative Density Propagation for Visual Tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 11, Nov. 2007.

[7] Z. Liu, J. Zhu, J. Bu, C. Chen. A Survey of Human Pose Estimation: The Body Parts Parsing based Methods. Journal of Visual Communication and Image Representation, Volume 32, October 2015, Pages 1019.

[8] A. Agarwal, B. Triggs. 3D Human Pose from Silhouettes by Relevance Vector Regression. CVPR 2004.

[9] G. Shakhnarovich, P.A. Viola, T. Darrell. Fast Pose Estimation with Parameter-Sensitive Hashing. ICCV 2003.

[10] Y. Hoshen, S. Peleg. An Egocentric Look at Video Photographer Identity. CVPR, 2016.

[11] C. Li, K.M. Kitani. Model Recommendation with Virtual Probes for Ego-Centric Hand Detection. ICCV 2013.

[12] C. Li and K.M. Kitani. Pixel-level Hand Detection for Egocentric Videos. CVPR 2013.

[13] X. Ren and C. Gu. Figure-ground Segmentation Improves Handled Object Recognition in Egocentric Video. CVPR 2010.

[14] A. Fathi, A. Farhadi, and J.M. Rehg. Understanding Egocentric Activities. ICCV 2011.

[15] Y. Li, Z. Ye, J.M. Rehg. Delving into Egocentric Actions. CVPR 2015.

[16] B. Kopp, A. Kunkel, H. Flor, T. Platz, U. Rose, K. Mauritz, K. Gresser, K. McCulloch, E. Taub. The Arm Motor Ability Test: reliability, validity, and sensitivity to change of an instrument for assessing disabilities in activities of daily living. Arch. of physical medicine and rehab, 78(6), 1997.

[17] Y. Poleg, C. Arora, and S. Peleg. Temporal Segmentation of Egocentric Videos. CVPR 2014.

[18] D. Gutierrez-G, L. Puig, J.J. Guerrero, Full Scaled 3D Visual Odometry from a Single Wearable Omnidirectional Camera. IROS 2012.

[19] D. Gutierrez-G, J.J. Guerrero. Scaled monocular SLAM for walking people. ISWC 2013.

[20] D. Damen, T. Leelasawassuk, O. Haines, A. Calway, W.W. Mayol-Cuevas. You-Do, I-Learn: Discovering Task Relevant Objects and their Modes of Interaction from Multi-User Egocentric Video. BMVC 2014

[21] H. Pirsiavash and D. Ramanan. Detecting Activities of Daily Living in First-Person Camera Views. CVPR 2012.

[22] B. Soran, A. Farhadi, L.G. Shapiro. Action Recognition in the Presence of One Egocentric and Multiple Static Cameras. ACCV 2014.

[23] K. Kitani, T. Okabe, Y. Sato, and A. Sugimoto. Fast Unsupervised Ego-Action Learning for First-Person Sports Video. CVPR 2011.

[24] E. Spriggs, F.D. la Torre, and M. Hebert. Temporal Segmentation and Activity Classification from First-Person Sensing. CVPR Workshop on Egocentric Vision, 2009.

[25] L. Kovar, M. Gleicher, F. Pighin. Motion Graphs. Siggraph 2002.

[26] A. Krizhevsky, I. Sutskever, G.E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. NIPS 2012.

[27] L.B. Smith, C. Yu, and A.F. Pereira. Not Your Mothers View: the Dynamics of Toddler Visual Experience. Developmental Science, Volume 14, No. 1, pp 917, 2011.

[28] R. Mur-Artal, J.M.M. Montiel and J.D. Tardós. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. IEEE Transactions on Robotics, vol. 31, no. 5, pp. 1147-1163, October 2015.

[29] H. Sidenbladh, M. J. Black, L. Sigal. Implicit Probabilistic Models of Human Motion for Synthesis and Tracking. ECCV 2002