MonoPerfCap: Human Performance Capture from Monocular Video

INTRODUCTION

- ➔ MonoPerfCap tackles the problem of human performance capture with general clothing from monocular RGB videos.
- ➔ It can work even outdoors, with general background.
- ➔ It overcomes the limitation of depth cameras and multi-view setups.

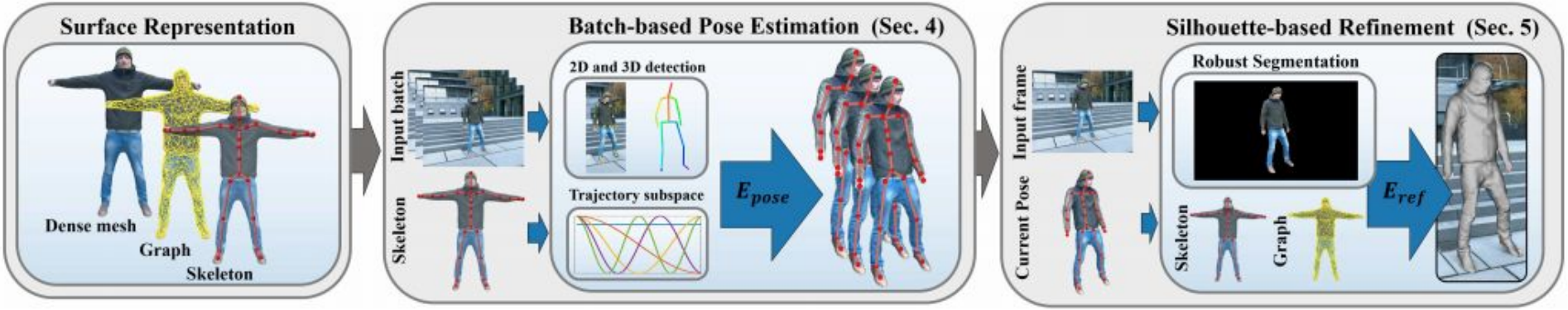
KEY ASPECTS OF THE METHOD

- ➔ Marker-less approach
- ➔ Temporally coherent
- ➔ Reconstructs articulated human skeleton motion
- ➔ Reconstructs non-rigid surface deformation
- ➔ Pre-acquired person-specific template mesh is needed

METHOD OVERVIEW

- Human motion is parameterized based on two level deformation hierarchy:
  - **Coarser level -** Articulated motion is captured in skeleton deformation space.
  - **Finer Level -** A deformation field is parameterized by an embedded deformation graph

- Motion capture is performed in a **coarse-to-fine** method, based on **two** steps:
  - **Batch based pose estimation**
  - **Silhouette based refinement**


Surface Representation
Dense mesh
Graph
Skeleton
Batch-based Pose Estimation (Sec. 4)
Input batch
Skeleton
2D and 3D detection
Trajectory subspace
E_{pose}
Silhouette-based Refinement (Sec. 5)
Input frame
Current Pose
Robust Segmentation
Skeleton
Graph
E_{ref}

TEMPLATE MESH ACQUISITION

- ➔ High res. video of the person standing in a static T-pose is recorded using a handheld camera orbiting around the person.
- ➔ 60 images are uniformly sampled from the video for image based reconstruction of triangulated surface.

BATCH BASED POSE ESTIMATION

Articulated human motion is parameterized based on low dimensional skeleton subspace $\mathbf{S = \{t, R, \Theta\}}$

$\mathbf{t}$ = Position of root joint $\mathbf{(t \in R^3)}$
$\mathbf{R}$ = Rotation of root joint $\mathbf{(R \in SO(3))}$
$\mathbf{\Theta}$ = 27 joint angles stacked together $\mathbf{(\Theta \in R^{27})}$

$\mathbf{N_d}$ = 16 (No. of joints)

This leads to a **33 dimensional** deformation subspace.

BATCH BASED POSE ESTIMATION CONTD.

- Given monocular input video $\mathcal{V} = \{I_f\}_{f=1}^{N}$, with N image frames I_f .
- The goal is to estimate skeleton parameters S_f for all input frames.
- A novel batch-based approach is proposed that jointly recovers the motion for a continuous window in time:

$$\mathcal{B} = \{S_f \mid f_{start} \leq f \leq f_{end}\}$$

f_{start} = index of the first frame of the batch
f_{end} = index of the last frame of the batch

$|\mathcal{B}| = 50$

- The input video is partitioned into a series of overlapping frames (10 frames overlap), and a linear blending function is used in the overlap region.

BATCH BASED POSE ESTIMATION CONTD.

The problem of estimating the articulated motion of each batch B is phrased as a constrained optimization problem-

$$\begin{aligned} \mathcal{B}^* = \underset{\mathcal{B}}{\text{argmin}} \quad & E_{\text{pose}}(\mathcal{B}) , \\ \text{subject to} \quad & \boldsymbol{\Theta}_{min} \leqslant \boldsymbol{\Theta}_f \leqslant \boldsymbol{\Theta}_{max} , \\ & \forall f \in [f_{start}, f_{end}] , \end{aligned}$$

Batch based pose estimation objective-

$$E_{\text{pose}}(\mathcal{B}) = \underbrace{E_{\text{2d}}(\mathcal{B}) + w_{3d} E_{\text{3d}}(\mathcal{B})}_{\text{data fitting}} + \underbrace{w_d E_{\text{d}}(\mathcal{B})}_{\text{regularization}}$$

BATCH BASED POSE ESTIMATION CONTD.

$\mathbf{E_{2d}}$ **=>** based on joint detections in image space

$\mathbf{E_{3d}}$ **=>** based on 3D joints

$\mathbf{E_{d}}$ **=>** based on DCT

For each input image $\boldsymbol{I_f}$ and each of the 16 joints $\boldsymbol{J_i}$, we estimate -

$\mathbf{d}_{f,i}^{2d}$ => 2D position of the joint in image space

$\mathbf{d}_{f,i}^{3d}$ => 3D position of the joint

Resnet based CNN joint position regression method of **[Mehta et al. 2016]** is used to determine these terms.

BATCH BASED POSE ESTIMATION CONTD.

- ➔ The **2D** pose network is trained on the **MPII Human Pose** and **LSP** datasets, and the **3D** pose network is fine-tuned from the 2D pose network on the **H3.6M** and **3DHP** datasets.

- ➔ 2D joint alignment term is a re-projection constraint enforcing that the projected joint positions $J_i(S_f)$ closely match the corresponding 2D detections:

$$E_{2d}(\mathcal{B}) = \frac{1}{|\mathcal{B}|} \sum_{S_f \in \mathcal{B}} \frac{1}{N_d} \sum_{i=1}^{N_d} \left\| \Pi(J_i(S_f)) - \mathbf{d}_{f,i}^{2d} \right\|_2^2$$

(**Π : $R^3 \rightarrow R^2$** implements the full perspective camera projection)

BATCH BASED POSE ESTIMATION CONTD.

$$E_{3d}(\mathcal{B}) = \frac{1}{|\mathcal{B}|} \sum_{\mathcal{S}_f \in \mathcal{B}} \frac{w_f}{N_d} \sum_{i=1}^{N_d} \left\| J_i(\mathcal{S}_f) - (\mathbf{d}_{f,i}^{3d} + \mathbf{t}_f) \right\|_2^2$$

- Since the 3D joint detections are normalized for a skeleton with average bone length and are predicted relative to the root joint, rather than in camera space, they have to be rescaled to match the actor model, and mapped to their corresponding camera space position based on an unknown per-frame global translation $\boldsymbol{t_f}$.

- In order to prune frames with low 3D detection confidence, we measure the per-frame PCK error $\boldsymbol{PCK_f}$ between the 2D joint detections and the projected 3D detections and apply a per-frame binary weight wf to the 3D data term.

$$w_f = \begin{cases} 1 & \text{if } \mathrm{PCK}_f < thres_{pck}, \\ 0 & \text{else.} \end{cases}$$

BATCH BASED POSE ESTIMATION CONTD.

- ➔ Temporal smoothness is imposed by forcing the trajectory of each skeleton parameter to lie on a low dimensional linear subspace.

- ➔ All pose estimates $S_f \in B$ are coupled by minimizing the distance to a K = 8 dimensional linear subspace $DCT \in R^{K \times |B|}$ **[Park et al. 2015]** spanned by the K lowest frequency basis vectors of the discrete cosine transform (DCT).

$$E_d(\mathcal{B}) = \frac{1}{|\mathcal{B}|} \left\| \Lambda \mathbf{S}_{\mathcal{B}} \, \mathrm{Null}(\mathbf{DCT}) \right\|_F^2$$

$$\mathbf{S}_{\mathcal{B}} = [\mathcal{S}_{f_{start}}, \quad \dots \quad, \mathcal{S}_{f_{end}}] \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{B}|}$$

$$\Lambda = \mathrm{diag}([\lambda_{\mathbf{t}}, \lambda_{\mathbf{R}}, \lambda_{\boldsymbol{\Theta}}])$$

- ➔ Finally, the constrained non-linear least squares optimization problem is solved using the Levenberg Marquardt (LM) algorithm.

SILHOUETTE-BASED REFINEMENT:

Why Refinement:

Batch-based pose optimization does not capture non-rigid surface deformation due to apparel and skin, and thus leads to misalignments between the skeleton-deformed template mesh and the input images, particularly at the boundaries.

This has 3 tasks

1)Automatic Silhouette Extraction

2)Silhouette-based Pose Refinement

SILHOUETTE-BASED REFINEMENT:

3)Silhouette-based Non-Rigid Surface Refinement

1)Automatic Silhouette Extraction:

To extract silhouette GrabCut was used.

GrabCut requires a user-specified initialization $T = \{T_b , T_{ub} , T_{uf} , T_f \}$.

T_b , T_f = known background and foreground masks.

T_{ub} , T_{uf} = uncertain background and foreground regions.

Segmentation is performed over T_{ub} , T_{uf} .

SILHOUETTE EXTRACTION:

To automate this we first rasterize the skeleton and the deformed dense actor template V (S_f) to obtain two masks R and M respectively.

T is initialised as following:

$T_f = R \cup \text{erosion}(M)$,

$T_b = \text{dilation}(M)$,

$T_{uf} = M - T_f$,

$T_{ub} = \text{dilation}(M) - M$.

SILHOUETTE EXTRACTION:

To improve robustness motion cues were incorporated in the objective. the temporal per-pixel color gradients between adjacent frames and encourage neighboring pixels with small temporal gradients to belong to the same region.

The masks $\{T_b , T_{ub} , T_{uf} , T_f\}$ are illustrated in red, blue, yellow and green.

2)Silhouette-based Pose Refinement

Refinement is performed in an Iterative Closest Point (ICP) manner.

In an iteration, for each boundary point of the projected surface model, search for its closest point on the image silhouette that shares a similar normal direction.

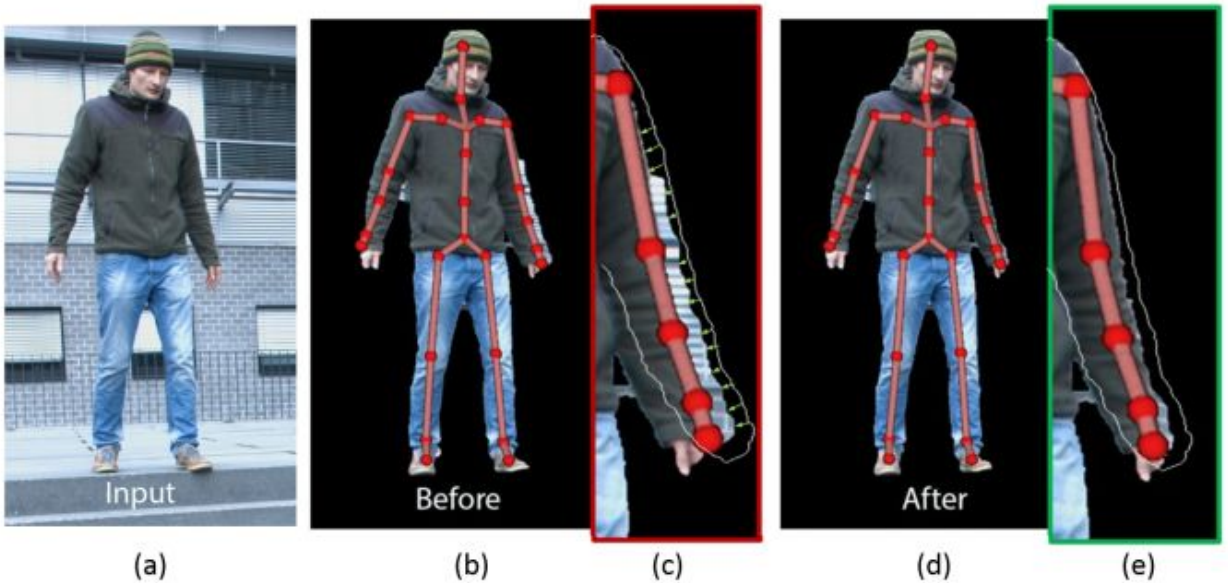
refine the pose by solving the following non-linear least squares optimization

$\mathbf{E_{ref}(S_f) = E_{con}(S_f) + w_{stab}E_{stab}(S_f)}$ where $\mathbf{E_{con}}$ aligns the mesh boundary with the input silhouette, $\mathbf{E_{stab}}$ constrains the solution to stay close to the batch-based results and w stab balances the importance of the two terms.

$\mathbf{E_{con}(S_f) = 1/|S| \Sigma[n^T_k.(\Pi(V_k(S_f) - s_k)]^2 \quad \forall\ k \in S}$ where S is the boundary of the actor model, v_k the position of vertex k and $s_k \in R^2$ the corresponding silhouette point in the image with 2D normal n_k .

Refinement Continued:

$\mathbf{E_{stab}(S_f)=1/N_d\,\Sigma||J_i(S_f)-J_i(S^*_f)||\ \forall\ i=1\ to\ N_d}$ where S^*_f are the joint angles after batch-based pose estimation and $J_i(\cdot)$ computes the 3D position of joint J_i.



3) Silhouette-based Non-Rigid Surface Refinement

A deformation graph D, consisting of M ≈ 1000 nodes, is generated from the template mesh using a uniform mesh decimation/simplification strategy.

A radius of influence is assigned for each deformation node by computing the maximum geodesic distance to its connected graph nodes.

Non-Rigid Surface Refinement Contd:

Each node defines a local warp field W_i that rotates $R_i \in SO(3)$ and translates $t_i \in R^3$ points $x \in R^3$ in the surrounding space.

$\mathbf{W_i(x) = R_i(x - \hat{g}_i) + \hat{g}_i + t_i}$ **,** where $\hat{g}_i \in R^3$ is the canonical position of node i, computed with the result of the pose refinement.

The graph and it's DoF are represented as $D = \{(R_i , t_i)| \, i \in [0, M)\}$.

The deformation is achieved by linear blending of the per-node warp fields i.e,

$\mathbf{v_i = W (\hat{v}_i) = \Sigma \, b_{i,k}(x) \cdot W_k (\hat{v}_i) \quad \forall \, k \in F_i}$ **,** where $v_i \in R^3$ is the deformed vertex position, $\hat{v}_i \in R^3$ is the canonical position of vertex i and $\mathbf{F_i}$ is the set of deformation nodes that influence vertex i.

Non-Rigid Surface Refinement Contd:

Given the embedded deformation graph, our silhouette-based surface refinement is expressed as the following optimization problem:

$\mathbf{E_{surf}(D) = E_{con}(D) + w_{arap}\, E_{arap}(D)}$. Here $\mathbf{E_{con}}$ is the silhouette alignment term, $\mathbf{E_{arap}}$ is as-rigid-as-possible regularization term.

$\mathbf{E_{con}(D) = 1/|S|\ \Sigma[n^T_k.(\Pi(V_k(D) - s_k)]^2\ \ \forall\ k \in S}$ where S is the model silhouette, v_k the position

of vertex k and $s_k \in R^2$ its corresponding silhouette point with normal $n_k \in R^2$.

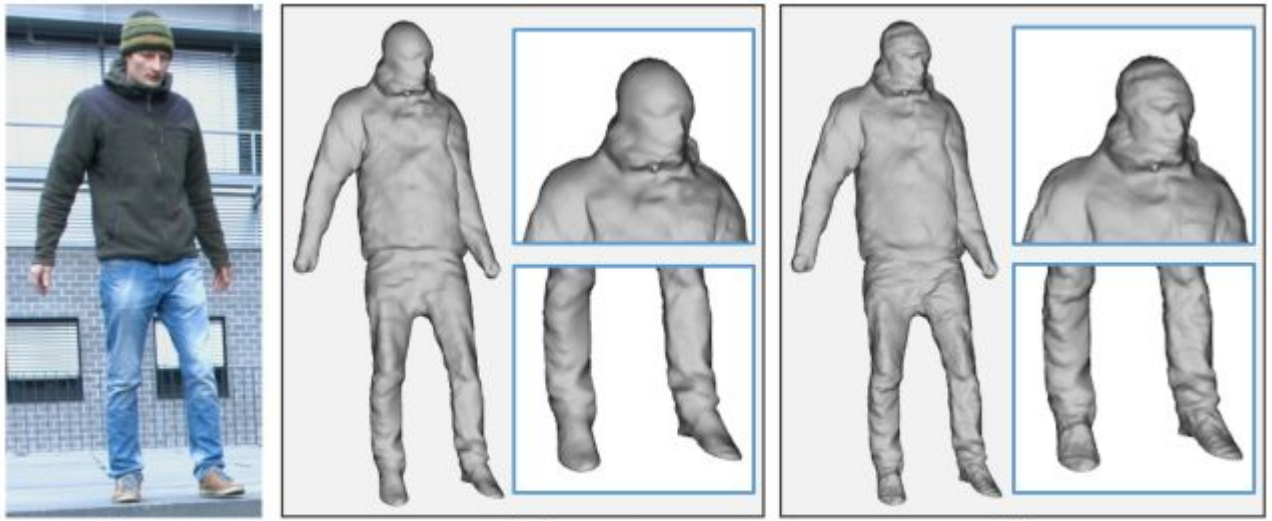
$\mathbf{E_{arap}(D)=1/M*\sum_{i=1}^{m} w_i \sum_{j \in N(i)} \| (g_i - g_j) - Ri(\hat{g}_i - \hat{g}_j) \|_2^2}$ Here, $g_i = W_i(\hat{g}\, i) = \hat{g}_i + t\, i$ is the deformed

position of node $\hat{g}_i$ and N_i is its 1-ring neighbourhood.

Contd.

Surface refinement is performed in an ICP like manner.

Initialize the optimization problem based on the pose refinement result and minimize E_{surf} using the LM algorithm.



Results from Paper