Supplementary Material (Full Body Performance Capture under Uncontrolled and Varying Illumination : A Shading-based Approach)

Chenglei Wu[1,2], Kiran Varanasi[1], and Christian Theobalt[1]

[1]Max Planck Institute for Informatik
[2]Intel Visual Computing Institute

Here we provide more elaborate derivations of the equations discussed in the main paper in sections 5.1 and 5.2. For better readability, we repeat the discussion in the paper of these two sections, but substantiate the equations now with the exact form of the functions and matrices. In the following, we cite the reference Bregler et al.[1] of the main paper at a few places. Please find this reference from the main paper. We will publish this document as a publicly available technical report after the acceptance of the main paper.

1 Surface parameterization with respect to pose

We use the popular linear blend skinning approach to deform the mesh to a skeletal pose. Given the position of vertex i to be q_i^t at time t, this vertex's new position q_i^{t+1} at time $t+1$ has the following form:

$$\begin{pmatrix} q_i^{t+1} \\ 1 \end{pmatrix} = \sum_{j=1}^{m} w_j C_{J_j} \begin{pmatrix} q_i^t \\ 1 \end{pmatrix}, \tag{1}$$

where C_{j_j} represents the rigid motion of joint (or bone) J_j. Each vertex i is assigned a set of skinning weights w_j that determine how much influence bone (or joint) j has on the deformation of vertex i.

Following Bregler et al.[1], we represent the articulated pose to be estimated by a set of twists $\theta_k \hat{\xi_k}$. The state of a kinematic chain is determined by a global twist $\hat{\xi}$ and the joint angles $\Theta = (\theta_1, \cdots, \theta_m)$. Assuming the state of the kinematic skeleton of the previous time-step to be known, the unknowns for pose estimation are the rigid motion of the root node and changes in joint angles which we denote as

$$\phi = (\Delta\hat{\xi}, \Delta\theta_1, \cdots, \Delta\theta_m) \tag{2}$$

Let q_i^t be the position of vertex i at t. By using exponential maps to represent each joint's rigid motion and by linearizing the rigid body transforms, the pose of the vertex i at $t+1$ can be expressed with the skinning equation as

$$\begin{aligned}\begin{pmatrix} q_i^{t+1} \\ 1 \end{pmatrix} &= \sum_{j=1}^{m} w_j e^{\Delta\hat{\xi}} \prod_{k \in T(j)} e^{\hat{\xi_k} \cdot \Delta\theta_k} \begin{pmatrix} q_i^t \\ 1 \end{pmatrix} \\ \approx \begin{pmatrix} q_i^t \\ 1 \end{pmatrix} &+ \left(\Delta\hat{\xi} + \sum_{j=1}^{m} w_j \sum_{k \in T(j)} \hat{\xi_k} \cdot \Delta\theta_k \right) \begin{pmatrix} q_i^t \\ 1 \end{pmatrix}, \end{aligned} \tag{3}$$

where $T(j)$ determines the indices of joints preceding the joint k in the kinematic chain. In another way, Eq. (3) can be rewritten as:

$$\begin{aligned}\begin{pmatrix} \Delta q_i^t \\ 0 \end{pmatrix} = \begin{pmatrix} q_{t+1} \\ 1 \end{pmatrix} - \begin{pmatrix} q_i^t \\ 1 \end{pmatrix} \approx \left(\Delta\hat{\xi} + \sum_{j=1}^{m} w_j \sum_{k \in T(j)} \hat{\xi_k} \cdot \theta_k \right) \begin{pmatrix} q_t \\ 1 \end{pmatrix} \\ = \left[\boldsymbol{I}_{4\times3}, -\hat{q_i^t}, W_{T(1)}\hat{\xi_1} \begin{pmatrix} q_i^t \\ 1 \end{pmatrix}, W_{T(2)}\hat{\xi_2} \begin{pmatrix} q_i^t \\ 1 \end{pmatrix}, \ldots, W_{T(m)}\hat{\xi_m} \begin{pmatrix} q_i^t \\ 1 \end{pmatrix} \right] \cdot \boldsymbol{\phi} = M_q(i) \cdot \phi, \end{aligned} \tag{4}$$

where $W_{T(j)} = \sum_{k \in T(j)} w_k$ and is the total sum of the skinning weights which are influenced by the joint j (i.e. corresponding to joints appearing after the joint j in the kinematic chain), and $M_q(i)$ is the matrix determining how the pose-change ϕ influences the change of vertex position. Thus, the change in each vertex position is expressed as a linear function of ϕ.

A similar equation can be derived for the vertex normal n_i^{t+1} at time $t+1$

$$\begin{aligned}\Delta n_i^{t+1} = (\Delta\hat{\omega} + \sum_{j=1}^{m} w_j \sum_{k \in T(j)} \hat{\omega_k} \cdot \theta_k) \cdot n_i^t \\ = \left[\mathbf{0}, -\hat{n_i^t}, W_{T(1)}\hat{\omega_1} n_i^t, W_{T(2)}\hat{\omega_2} n_i^t, \ldots, W_{T(m)}\hat{\omega_m} n_i^t \right] \cdot \boldsymbol{\phi} = M_n(i) \cdot \phi, \end{aligned} \tag{5}$$

where $\hat{\omega}$ is the rotation part of twist $\hat{\xi}$, and $M_n(i)$ is a matrix that determines how the pose-change ϕ results in a change in normal orientation.

2 Shading constraint for pose estimation

Our shading constraint requires the rendered images of the optimal pose according to our lighting model to be as-close-as-possible to the image data captured. The shading constraint for a single camera c is defined as

$$E_c^s = \sum_i (\rho_i g(q_i^{t+1}) \cdot S(n_i^{t+1}) - I_c^{t+1}(x_i^{t+1}, y_i^{t+1}))^2, \tag{6}$$

where (x_i^{t+1}, y_i^{t+1}) is the projection of the surface vertex q_i^{t+1}, and $g(q_i^{t+1})$ and $S(n_i^{t+1})$ are the vectors of SH coefficients g_k and S_k (See the image formation model in the Eq. (2) of the paper for description of these symbols). Next we linearize the SH term $S(n_i^{t+1})$ and the image intensity term I_c^{t+1}. The SH term is expressed in a first-order Taylor-series expansion, and using the terms of Eq. (5).

$$S(n_i^{t+1}) \approx S(n_i^t) + \frac{\partial S(n_i^t)}{\partial n_i^t} \Delta n_i^t = S(n_i^t) + \frac{\partial S(n_i^t)}{\partial n_i^t} M_n(i) \cdot \boldsymbol{\phi}, \tag{7}$$

where $\frac{\partial S(n_i^t)}{\partial n_i^t}$ is derivative of scaled SH function with respect to normal changes Δn_i^t, which are expressed in terms of pose changes ϕ. See the appendix **??** for the detailed form of $S(n_i^t)$ and $\frac{\partial S(n_i^t)}{\partial n_i^t}$.

Inspired by the formulation of optical flow, we linearize $I^{t+1}(x_i^{t+1}, y_i^{t+1})$ as:

$$I^{t+1}(x_i^{t+1}, y_i^{t+1}) = I^{t+1}(x_i^t + u_i, y_i^t + v_i) \approx I^{t+1}(x_i^t, y_i^t) + \nabla_x I^{t+1} u_i + \nabla_y I^{t+1} v_i. \tag{8}$$

Next, we derive the linear approximation for the flow (u_i, v_i) in an image from the motion parameters ϕ. As camera calibration is available in our system, we use the full perspective camera model instead of scaled orthographic projection as used by Bregler et al.[1]. The full perspective camera model has the following form:

$$\begin{pmatrix} x_i^t \\ y_i^t \end{pmatrix} = \begin{pmatrix} \frac{s_1}{Z_i^{t+1}} & 0 & 0 \; s_3 \\ 0 & \frac{s_2}{Z_i^{t+1}} & 0 \; s_4 \end{pmatrix} \cdot \boldsymbol{e}^{\hat{\xi_c}} \cdot \begin{pmatrix} q_i^{t+1} \\ 1 \end{pmatrix}, \tag{9}$$

where s_1, s_2, s_3, s_4 are the acquired camera intrinsic parameters, Z_i^{t+1} is the depth of q_i^{t+1} for the current camera, and $\boldsymbol{e}^{\hat{\xi_c}}$ acts as the extrinsic matrix of the camera's pose. Then, the image motion from time t to time $t+1$ can be linearized as:

$$\begin{aligned}
\begin{pmatrix} u_i \\ v_i \end{pmatrix} &= \begin{pmatrix} \frac{s_1}{Z_i^{t+1}} & 0 & 0 \; s_3 \\ 0 & \frac{s_2}{Z_i^{t+1}} & 0 \; s_4 \end{pmatrix} \cdot \boldsymbol{e}^{\hat{\xi_c}} \cdot \begin{pmatrix} q_i^{t+1} \\ 1 \end{pmatrix} - \begin{pmatrix} \frac{s_1}{Z_i^t} & 0 & 0 \; s_3 \\ 0 & \frac{s_2}{Z_i^t} & 0 \; s_4 \end{pmatrix} \cdot \boldsymbol{e}^{\hat{\xi_c}} \cdot \begin{pmatrix} q_i^t \\ 1 \end{pmatrix} \\
&= \begin{pmatrix} \frac{s_1}{Z_i^t - \Delta Z_i^t} & 0 & 0 \; s_3 \\ 0 & \frac{s_2}{Z_i^t - \Delta Z_i^t} & 0 \; s_4 \end{pmatrix} \cdot \boldsymbol{e}^{\hat{\xi_c}} \cdot \begin{pmatrix} q_i^{t+1} \\ 1 \end{pmatrix} - \begin{pmatrix} \frac{s_1}{Z_i^t} & 0 & 0 \; s_3 \\ 0 & \frac{s_2}{Z_i^t} & 0 \; s_4 \end{pmatrix} \cdot \boldsymbol{e}^{\hat{\xi_c}} \cdot \begin{pmatrix} q_i^t \\ 1 \end{pmatrix} \\
&\approx \begin{pmatrix} \frac{s_1}{Z_i^t} & 0 & 0 \; s_3 \\ 0 & \frac{s_2}{Z_i^t} & 0 \; s_4 \end{pmatrix} \cdot \boldsymbol{e}^{\hat{\xi_c}} \cdot \left(\begin{pmatrix} q_i^{t+1} \\ 1 \end{pmatrix} - \begin{pmatrix} q_i^t \\ 1 \end{pmatrix} \right) + \begin{pmatrix} \frac{s_1 \Delta Z_i^t}{{Z_i^t}^2} & 0 & 0 \; 0 \\ 0 & \frac{s_2 \Delta Z_i^t}{{Z_i^t}^2} & 0 \; 0 \end{pmatrix} \cdot \boldsymbol{e}^{\hat{\xi_c}} \cdot \begin{pmatrix} q_i^{t+1} \\ 1 \end{pmatrix} \\
&\approx \begin{pmatrix} \frac{s_1}{Z_i^t} & 0 & 0 \; s_3 \\ 0 & \frac{s_2}{Z^t} & 0 \; s_4 \end{pmatrix} \cdot \boldsymbol{e}^{\hat{\xi_c}} \cdot \begin{pmatrix} \Delta q_i^t \\ 0 \end{pmatrix} + \begin{pmatrix} \frac{s_1 \Delta Z_i^t}{{Z_i^t}^2} & 0 & 0 \; 0 \\ 0 & \frac{s_2 \Delta Z_i^t}{{Z_i^t}^2} & 0 \; 0 \end{pmatrix} \cdot \boldsymbol{e}^{\hat{\xi_c}} \cdot \left(\begin{pmatrix} q_i^t \\ 1 \end{pmatrix} + \begin{pmatrix} \Delta q_i^t \\ 0 \end{pmatrix} \right) \\
&\approx \begin{pmatrix} \frac{s_1}{Z_i^t} & 0 & 0 \; s_3 \\ 0 & \frac{s_2}{Z_i^t} & 0 \; s_4 \end{pmatrix} \cdot \boldsymbol{e}^{\hat{\xi_c}} \cdot \begin{pmatrix} \Delta q_i^t \\ 0 \end{pmatrix} + \begin{pmatrix} \frac{s_1}{{Z_i^t}^2} & 0 & 0 \; 0 \\ 0 & \frac{s_2}{{Z_i^t}^2} & 0 \; 0 \end{pmatrix} \cdot \boldsymbol{e}^{\hat{\xi_c}} \cdot \begin{pmatrix} q_i^t \\ 1 \end{pmatrix} \cdot \Delta Z_i^t,
\end{aligned} \tag{10}$$

where Z_i^t is the depth of q_i^t for the current camera. The linearization is based on the assumption that the rigid motion Δq_i^t as well as the relative depth change

$\Delta Z_i/Z_i^t$ are small enough. Besides, the depth change ΔZ_i^t can be further expressed through the motion parameters:

$$\begin{aligned}\Delta Z_i^t &= -\left[\boldsymbol{e}^{\hat{\xi_c}}\begin{pmatrix}\Delta q_i^t\\ 0\end{pmatrix}\right]_z = -\left[\boldsymbol{e}^{\hat{\xi_c}}\cdot M_q(i)\cdot\phi\right]_z\\ &= -\left[\begin{pmatrix}r_1^T\ t_1\\ r_2^T\ t_2\\ r_3^T\ t_3\end{pmatrix}\cdot M_q(i)\cdot\phi\right]_z = -\left[r_3^T\ t_3\right]\cdot M_q(i)\cdot\phi,\end{aligned} \tag{11}$$

where r_3^T is the $3rd$ row of the rotation matrix of the camera pose. As the $4th$ row of $M_q(i)$ is zeros, t_3 can be omitted in the above equation. So the flow (u_i, v_i) can ultimately be expressed as a linear function of the pose change ϕ as following:

$$\begin{aligned}\begin{pmatrix}u_i\\ v_i\end{pmatrix} &\approx \begin{pmatrix}\frac{s_1}{Z_i^t} & 0 & 0 & s_3\\ 0 & \frac{s_2}{Z_i^t} & 0 & s_4\end{pmatrix}\cdot\boldsymbol{e}^{\hat{\xi_c}}\cdot\begin{pmatrix}\Delta q_i^t\\ 0\end{pmatrix} + \begin{pmatrix}\frac{s_1}{Z_i^{t2}} & 0 & 0 & 0\\ 0 & \frac{s_2}{Z_i^{t2}} & 0 & 0\end{pmatrix}\cdot\boldsymbol{e}^{\hat{\xi_c}}\cdot\begin{pmatrix}q_i^t\\ 1\end{pmatrix}\cdot\Delta Z_i^t\\ &\approx \left\{\begin{pmatrix}\frac{s_1}{Z_i^t} & 0 & 0 & s_3\\ 0 & \frac{s_2}{Z_i^t} & 0 & s_4\end{pmatrix}\boldsymbol{e}^{\hat{\xi_c}} - \begin{pmatrix}\frac{s_1}{Z_i^{t2}} & 0 & 0 & 0\\ 0 & \frac{s_2}{Z_i^{t2}} & 0 & 0\end{pmatrix}\boldsymbol{e}^{\hat{\xi_c}}\begin{bmatrix}q_i^t\\ 1\end{bmatrix}\cdot\left[r_3^T\ 0\right]\right\}\cdot M_q(i)\cdot\phi,\end{aligned} \tag{12}$$

Thus, the flow (u_i, v_i) is ultimately be expressed as a linear function of pose-change parameters ϕ.

The shading constraint in Eq. (6) can be further improved by considering the color similarity between the rendered color and the image color. This color similarity is computed as the Euclidean distance in HSV space and appears as a weighting factor α_i in our shading constraint. This helps us avoid optimizing the model where the template material does not yet match to its projection in the image. Combining terms from multiple cameras, our non-linear multi-view shading energy function is then given as

$$E = \frac{1}{N}\sum_c\sum_i\{\alpha_i^c(\rho_i g(q_i^{t+1})\cdot S(n_i^{t+1}) - I_c^{t+1}(x_i^{t+1}, y_i^{t+1}))\}^2, \tag{13}$$

where N is the total number of constraints for error normalization (*i.e,* the number of pixels in all cameras getting the projection from the mesh), and α_i^c is the color similarity for pixel i in camera c. Using the previously described recipe of linearization, this can be expressed in terms of pose parameters ϕ as a linear system:

$$\boldsymbol{H}\cdot\phi = \boldsymbol{b}. \tag{14}$$

Here, the k^{th} rows of matrix $\boldsymbol{H}$ and vector $\boldsymbol{b}$ have the following form:

$$\begin{aligned}\boldsymbol{H}_k = \alpha_i^c\rho_i g(q_i^{t+1})\cdot\frac{\partial S(n_i^t)}{\partial n_i^t}M_n(i) - \alpha_i^c\left[\tfrac{s_1}{Z_i^t}I_x^{t+1},\ \tfrac{s_2}{Z_i^t}I_y^{t+1},\ 0,\ s_3 I_x^{t+1} + s_4 I_y^{t+1}\right]\boldsymbol{e}^{\hat{\xi_c}}M_q(i)\\ +\alpha_i^c\left[\tfrac{s_1}{Z_i^{t2}}I_x^{t+1},\ \tfrac{s_1}{Z_i^{t2}}I_y^{t+1},\ 0,\ 0\right]\boldsymbol{e}^{\hat{\xi_c}}\begin{bmatrix}q_i^t\\ 1\end{bmatrix}\cdot\left[r_3^T\ 0\right]\cdot M_q(i),\\ \boldsymbol{b}_k = \alpha_i^c I^{t+1}(x_i^t, y_i^t) - \alpha_i^c\rho_i g(q_i^{t+1})\cdot S(n_i^t).\end{aligned} \tag{15}$$