LaTeX Author Guidelines for CVPR Proceedings

Anonymous CVPR submission

Paper ID ****

Abstract

The ABSTRACT is to be in fully-justified italicized text, at the top of the left-hand column, below the author and affiliation information. Use the word "Abstract" as the title, in 12-point Times, boldface type, centered relative to the column, initially capitalized. The abstract is to be in 10-point, single-spaced type. Leave two blank lines after the Abstract, then begin the main text. Look at previous CVPR abstracts to get a feel for style and length.

1. Settings

A hand in motion can be described by two types of parameters.

- Pose parameters *(joint angles and rigid rotation and translation of the hand)* $\theta \in \mathbb{R}^{N_\theta}$ change with time.
- Shape parameters *(length and thickness of fingers phalanges, position of finger bases, shape of the palm)* $\beta \in \mathbb{R}^{N_\beta}$ vary between different people while staying constant for each person.

Given a sequence of depth sensor images $\mathcal{D}_n$, the goal of our system is to get best possible estimate of shape parameters $\hat{\beta}_n$ from the input available so far.

How much information do we need to have a good guess of β? A trivial lower bound would finding β from a single sensor image. This scenario could be conceived for palm shape and fingers thickness, since they are "visible" in any hand pose. However, the phalanges length cannot be estimated when the fingers are straight; moreover the locations of finger bases are hidden under the skin and can only be inferred after seeing a range of hand poses. Sensor noise and $2,5D$ nature of the data further aggravate the problem. Previous authors [3] and [6] suggested to find hand shape from a set of manually picked hand poses.

Advantages of online optimization over batch optimization

- *user experience:* the user does not have to track with uncalibrated model first, manually pick the poses, wait till batch optimization finishes; also there is an immediate feedback on the result quality, so no need for several "blind" trials.
- *system efficiency:* there is not need to solve a potentially very big linear system, thus hand tracking algorithm will use less computational resources; potentially it can run on a lighter hardware.
- *result quality:* this claim will be /*hopefully*/ proven experimentally.
  - By default performance of batch algorithm is an upper bound for online algorithm running on the same data. However, in practice online algorithm can use any amount of data and batch algorithm is limited by the available memory and computational power.
  - Also, it is important that to take into account that the problem is solved with local optimization. If each frame of batch optimization is initialized far away from the true hand parameters, the solver can get stack in bad local optimum. (Shape prior should help in this case, but a good prior is difficult to learn). Online optimization, if it works properly, can have better and better initialization with every frame.

1.1. Hand Model

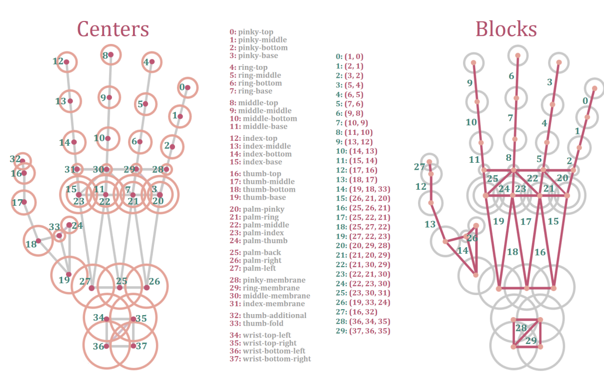

Figure 1. Sphere-mesh representation

CVPR
#****

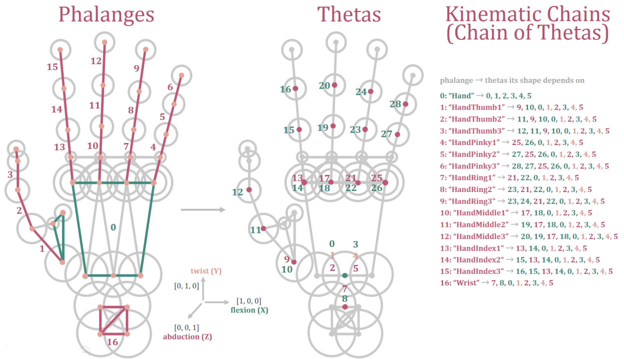

Figure 2. Pose parameters θ

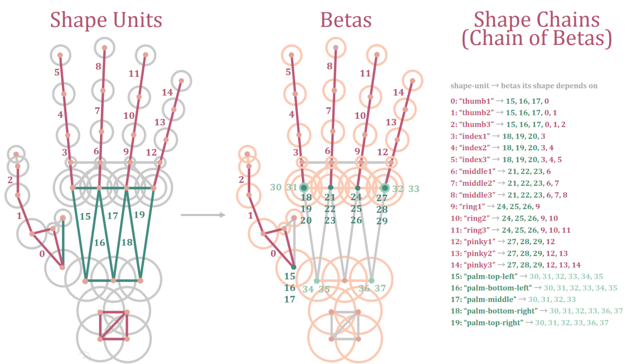

Figure 3. Shape parameters β

We use sphere-mesh hand model presented in [6], which is a special case of a convolution surface. **Maybe briefly recap advantages of sphere-meshes hand model representation**.

The skeleton $\mathcal{S} = \{\mathcal{V}, \mathcal{E}, \mathcal{F}\}$ of the sphere-mesh is a planar graph with a fixed topology. Each vertex $\mathcal{V}_i$ corresponds to a sphere with center $c_i \in \mathbb{R}^3$ and radius $r_i \in \mathbb{R}^+$. The radii at intermediate points are given by linear interpolation of the radii at the vertices. While the skeleton topology is predefined for the class of object "human hand", changing centers and radii spans the pose and shape space of human hands.

Note that shape parameters β stay constant for a given person and play a different semantic role from pose parameters θ, thus it is helpful to distinguish between to two types of the parameters. Also, it is important to prohibit the degrees of freedom that are not available to the human hand. Thus we use the two representations interchangeably - sphere-meshes representation for closest point correspondences and for rendering and "semantic" representation for finding β and θ. The centers c_i are computed from β and θ using forward kinematics with β being transitional degrees of freedom and θ - rotational degrees of freedom. The "impossible" degrees of freedom are eliminated by only allowing only biologically plausible rotational DOFs at the joints.

1.2. Parametrization

A point on the model surface m_i is defined by its barycentric coordinates $\{\kappa_1, \kappa_2, \kappa_3\}$ of its projection s_i on the corresponding skeleton face $\mathcal{F}_j = \{c_{j1}, c_{j2}, c_{j3}\}$ and the offset direction $o_i \in \mathbb{R}^3$

$$o_i = G_j^{-1} \frac{m_i - s_i}{\|m_i - s_i\|_2^2} \tag{1}$$

where G_j is global transformation of the face F_j given by forward kinematics.

Given the new parameters $\{\beta', \theta'\}$, the new position of the point is

$$\begin{aligned} m_i'(\beta', \theta') = \kappa_1 c_{j1}' + \kappa_2 c_{j1}' + \kappa_3 c_{j3}' + \\ + G_j'(\kappa_1 r_{j1}' + \kappa_2 r_{j2}' + \kappa_3 r_{j3}') o_i \end{aligned} \tag{2}$$

with $c_{j1}' = c_{j1}'(\beta', \theta')$, $r_{j1}' = r_{j1}'(\beta', \theta')$, $G_j' = G_j'(\theta')$.

1.3. Independent Solve

Denote the vector of all hand parameters as $x_n = [\theta_n; \beta_n]$ and the segmented hand point cloud in vectorized form as d_n.

Given d_n and initialization of hand pose from previous frame $x_n^0 = x_{n-1}^*$ we find an independent estimate of x_n^* by local Levenberg-Marquardt optimization of an objective similar to [6]

$$x_n^* = \text{argmin}_{x_n} \sum_{\tau \in \mathcal{T}} E_\tau(d_n, x_n) \tag{3}$$

where the energy terms in the objective function are the following:

- **d2m** model should be close to data
- **m2d** data should be close to model
- **limits** joint limits should hold
- **collision** fingers should not interpenetrate
- **shape** model shape should be plausible
- **pose** model pose should be plausible

See [6] for more details about energy terms. **Explain shape term somewhere later.**

1.4. Posterior distribution of the parameters

All the energy term $E_\tau(d_n, x_n)$ can be rewritten in a form $\|I_\tau d_n - F_\tau(x_n)\|_2^2$ with I_{d2m} an identity matrix, I_{m2d} a permutation(/reduction?) matrix that removes from d_n the entries that are not the closest for any model point and the remaining I_τ all zeros matrices. Denote $\left[F_{d2m}^T(x_n), ..., F_{pose}^T(x_n)\right]^T$ as $F(x_n)$ and overload d_n as $\left[(I_{d2m} d_n)^T, ..., (I_{pose} d_n)^T\right]^T$ for brevity of notation.

Given the hand point cloud at the first frame d_1, we find the best-fitting parameters (see Section 1.3)

$$x_1^* = \text{argmin}_{x_1} \underbrace{\log P(d_1|x_1)}_{L(x_1)} \tag{4}$$

with $P(d_1|x_1) = \exp\left(-(d_1 - F(x_1))^T(d_1 - F(x_1))\right)$. Expanding the log-likelihood of the data $L(x_1)$ around x_1^* we get

$$\begin{aligned} L(x_1) \approx L(x_1^*) + &\underbrace{\frac{\partial L}{\partial x_1}(x_1^*)}_{=0 \text{ at extremum}} (x_1 - x_1^*) - \\ -0.5(x_1 - x_1^*)^T &\underbrace{\frac{\partial\partial L}{\partial x_1 \partial x_1}(x_1^*)}_{\text{positive definite}}(x_1 - x_1^*) \end{aligned} \tag{5}$$

with $\frac{\partial L}{\partial x_1}(x_1^*) = -2F(x_1^*)^T + 2F(x_1)^T \frac{\partial F}{\partial x_1}(x_1^*)$ and $\frac{\partial\partial L}{\partial x_1 \partial x_1}(x_1^*) = 2\frac{\partial F}{\partial x_1}(x_1^*)^T \frac{\partial F}{\partial x_1}(x_1^*) + 2F(x_1^*)^T \frac{\partial\partial F}{\partial\partial x_1}(x_1^*) \approx \frac{\partial F}{\partial x_1}(x_1^*)^T \frac{\partial F}{\partial x_1}(x_1^*)$. Denoting $\frac{\partial\partial L^{-1}}{\partial x_n \partial x_n}(x_n^*)$ as Σ_n and taking an exponential of both sides gives:

$$P(d_1|x_1) \propto \exp\left(-0.5(x_1 - x_1^*)^T \Sigma_1^{-1}(x_1 - x_1^*)\right) \tag{6}$$

Thus, after processing the information from the first frame d_1, the quadratic approximation of posterior distribution of hand parameters is a normal distribution $\mathcal{N}\left(\hat{x}_1 = x_1^*, \hat{\Sigma}_1 = \Sigma_1\right)$ [ADD REFERENCE].

1.5. Combining independent measurements

The shape prior should only be used in the first frame, because afterwards it will be incorporated inside of Kalman prior.

Given the second frame d_2, we compute the corresponding value of the parameters x_2^* and their variance Σ_2 the same way as in 1.4. The posterior distribution of the parameters now has the form

$$\mathcal{N}(\hat{x}_2, \hat{\Sigma}_2) = \mathcal{N}(\hat{x}_1, \hat{\Sigma}_1)\mathcal{N}(x_2^*, \Sigma_2) \tag{7}$$

Applying the product of Gaussians rule presented in [5], we get

$$\begin{aligned} \hat{x}_2 = \Sigma_2(\hat{\Sigma}_1 + \Sigma_2)^{-1}\hat{x}_1 + \hat{\Sigma}_1(\hat{\Sigma}_1 + \Sigma_2)^{-1}x_2^* \\ \hat{\Sigma}_2 = \hat{\Sigma}_1(\hat{\Sigma}_1 + \Sigma_2)^{-1}\Sigma_2 \end{aligned} \tag{8}$$

In Appendix A it is shown that the Equations 8 are equivalent to Kalman filter update with measurement x_n^* and measurement noise covariance Σ_n. This is important, because, according to P. Maybeck [4], under the assumptions presented in Appendix A "... the Kalman filter can be shown to be the best filter of any conceivable form. It incorporates all information that can be provided to it. "

Table 1. KF: Combining independent measurements

$$\begin{gathered} x_n^* = \text{argmax}_{x_n} \log P(d_n|x_n) \\ \hat{x}_n = \text{argmax}_{x_n} \underbrace{\log\left(P(x_n|x_n^*)P(x_n|\hat{x}_{n-1})\right)}_{L(x_n)} \\ \text{with} \\ P(x_n|x_n^*) = \exp\left(-(x_n - x_n^*)^T \Sigma_n^{-1}(x_n - x_n^*)\right) \\ P(x_n|\hat{x}_{n-1}) = \exp\left(-(x_n - \hat{x}_{n-1})^T \hat{\Sigma}_{n-1}^{-1}(x_n - \hat{x}_{n-1})\right) \\ \hat{\Sigma}_n^{-1} = \frac{\partial\partial L}{\partial x_n \partial x_n}(\hat{x}_n) \approx \\ \begin{bmatrix} \left(\Sigma_n^{-1}\right)^{1/2} \\ \left(\hat{\Sigma}_{n-1}^{-1}\right)^{1/2} \end{bmatrix}^T \begin{bmatrix} \left(\Sigma_n^{-1}\right)^{1/2} \\ \left(\hat{\Sigma}_{n-1}^{-1}\right)^{1/2} \end{bmatrix} = \hat{\Sigma}_{n-1}^{-1} + \Sigma_n^{-1} \end{gathered}$$

1.6. Regularizing optimization

Note that the best-fitting parameters x_n^* for the given the single input frame d_n are computed using local LM optimization. Thus, it is crucial that optimization starts in a sensible region of parameters space. The optimization presented in Table 1 does not provided any information about the current estimate of the parameters $\hat{x}_n$ to the independent solve from Section 1.3. One way to remedy this is include the term $P(x_n|\hat{x}_{n-1})$ directly in the independent solve.

Table 2. Objective function of IEKF

$$\begin{gathered} \hat{x}_n = \text{argmax}_{x_n} \underbrace{\log\left(P(d_n|x_n)P(x_n|\hat{x}_{n-1})\right)}_{L(x_n)} \\ \text{with} \\ P(d_n|x_n) = \exp\left(-(d_n - F(x_n))^T(d_n - F(x_n))\right) \\ P(x_n|\hat{x}_{n-1}) = \exp\left(-(x_n - \hat{x}_{n-1})^T \hat{\Sigma}_{n-1}^{-1}(x_n - \hat{x}_{n-1})\right) \\ \hat{\Sigma}_n^{-1} = \frac{\partial\partial L}{\partial x_n \partial x_n}(\hat{x}_n) \approx \\ \begin{bmatrix} -\frac{\partial F}{\partial x_n}(\hat{x}_n) \\ \left(\hat{\Sigma}_{n-1}^{-1}\right)^{1/2} \end{bmatrix}^T \begin{bmatrix} -\frac{\partial F}{\partial x_n}(\hat{x}_n) \\ \left(\hat{\Sigma}_{n-1}^{-1}\right)^{1/2} \end{bmatrix} = \\ = \hat{\Sigma}_{n-1}^{-1} + \frac{\partial F}{\partial x_n}(\hat{x}_n)^T \frac{\partial F}{\partial x_n}(\hat{x}_n) = \hat{\Sigma}_{n-1}^{-1} + \Sigma_n^{-1} \end{gathered}$$

In Appendix D we demonstrate that optimizing the objective function presented in Table 2 using Levenberg-Marquardt algorithm is equivalent to measurement update of Iterated Extended Kalman Filter.

1.7. Comparing the two optimizations

The optimization presented in Table 1 will be used as a baseline. It treats the results of the independent solve x_n^* as a measurement, which, arguably can be considered as the simplest way to maintain an online estimate of the parameters. The optimization presented in Table 2 represents the hand tracking system as an Iterated Extended Kalman Filter. It directly outputs the estimated value of the parameters

$\hat{x}_n$, using d_n as a measurement.

A. Kalman Filter

Using the definitions from [7], Kalman Filter (KF) estimates the state $x_n \in \mathbb{R}^N$ of a linear system driven by the below equations, given the measurement $z_n \in \mathbb{R}^M$:

$$x_n = Ax_{n-1} + w_{n-1} \tag{9}$$
$$z_n = Jx_n + v_n \tag{10}$$

where w_n is process noise, v_n is measurement noise with $p(w) \sim \mathcal{N}(0, Q)$ and $p(v) \sim \mathcal{N}(0, R)$. The matrix A maps the state at the previous time step to the state at current time step. The matrix J maps the state x_n to the measurement z_n.

We assume that while there is no new measurements, an estimate of the current hand parameters are the previous known parameters up to Gaussian noise, that is $x_n = x_{n-1} + w_{n-1}$. We also assume for now that depth sensor noise is normally distributed, which actually is not the case. Thus, the states of our system can be approximated as following:

$$x_n = x_{n-1} + w_{n-1} \tag{11}$$
$$z_n = Jx_n + v_n \tag{12}$$

Define $\hat{x}_n^0$ to be initial state estimate at step n and $\hat{x}_n$ to be a state estimate after updating on the measurement z_n.

Then the covariance before and after the measurement is then

$$P_n^0 = \mathbb{E}[(x_n - \hat{x}_n^0)^T (x_n - \hat{x}_n^0)] \tag{13}$$
$$P_n = \mathbb{E}[(x_n - \hat{x}_n)^T (x_n - \hat{x}_n)] \tag{14}$$

Table 3. Time and measurement update for KF with A equal to identity

Time Update	Measurement Update
$\hat{x}_n^0 = \hat{x}_{n-1}$ $P_n^0 = P_{n-1} + Q$	$K_n = P_n^0 J^T (J P_n^0 J^T + R)^{-1}$ $\hat{x}_n = \hat{x}_n^0 + K_n(z_n - J\hat{x}_n^0)$ $P_n = (I - K_n J) P_n^0$

A.1. Special case from the Section 1.5

Let us consider the special case when the measurement $z_n = x_n^*$ is in the same space at the estimated state $\hat{x}_n$, thus J is equal to identity.

$$\begin{aligned}
\hat{x}_n &= \hat{x}_n^0 + P_n^0 (P_n^0 + R)^{-1} (z_n - \hat{x}_n^0) = \\
&= (P_n^0 + R)(P_n^0 + R)^{-1} \hat{x}_n^0 + P_n^0 (P_n^0 + R)^{-1} (z_n - \hat{x}_n^0) = \\
&= R (P_n^0 + R)^{-1} \hat{x}_n^0 + P_n^0 (P_n^0 + R)^{-1} z_n
\end{aligned}$$

$$\begin{aligned}
P_n &= ((P_n^0 + R)(P_n^0 + R)^{-1} - P_n^0 (P_n^0 + R)^{-1}) P_n^0 \\
P_n &= R (P_n^0 + R)^{-1} P_n^0
\end{aligned}$$

The expressions for $\hat{x}_n$ and P_n coincide with Equations 8 for product of two Gaussians with $z_n = x_n^*$, $P_n^0 = \hat{\Sigma}_{n-1}$ and $R = \Sigma_n$.

B. Extended Kalman Filter

Following [7], Extended Kalman Filter (EKF) estimates the state x_n of a non-linear system given the measurement z_n.

$$x_n = \tilde{F}(x_{n-1}, w_{n-1}) \tag{15}$$
$$z_n = F(x_n, v_n) \tag{16}$$

The function $F(\cdot)$ that maps the state x_n to the measurement z_n applies shape and pose parameters to the hand model and computes the closest model points to sensor data points. The function $\tilde{F}(\cdot)$ relates the state at the previous time step to the state at current time step, in our case $\tilde{F}(\cdot)$ is an identity mapping. Thus, $\frac{\partial \tilde{F}_{[i]}}{\partial x_{[j]}}(\hat{x}_{n-1}, 0) \equiv I$, $\frac{\partial \tilde{F}_{[i]}}{\partial w_{[j]}}(\hat{x}_{n-1}, 0) \equiv I$ and $\frac{\partial F_{[i]}}{\partial v_{[j]}}(\hat{x}_n^0, 0) \equiv I$, where I is an identity matrix of the corresponding size.

$$x_n = x_{n-1} + w_{n-1} \tag{17}$$
$$z_n = F(x_n) + v_n \tag{18}$$

Table 4. Time and measurement update for EKF with $\tilde{F}(\cdot)$ being an identity mapping

Time Update	Measurement Update
$\hat{x}_n^0 = \hat{x}_{n-1}$ $P_n^0 = P_{n-1} + Q$	$K_n = P_n^0 J_n^T (J_n P_n^0 J_n^T + R)^{-1}$ $\hat{x}_n = \hat{x}_n^0 + K_n(z_n - F_n)$ $P_n = (I - K_n J_n) P_n^0$

with $F_n = F(\hat{x}_n^0)$ and $J_{n\,[i,j]} = \frac{\partial F_{[i]}}{\partial x_{[j]}}(\hat{x}_n^0)$.

C. Iterated Extended Kalman Filter

In our case the measurement function $F(\cdot)$ is nonlinear such that the Levenberg-Marquardt algorithm $\hat{x}_n^{i+1} = (J(\hat{x}_n^i)^T J(\hat{x}_n^i) + \lambda I)^{-1} J(\hat{x}_n^i)^T F(\hat{x}_n^i)$ takes several iterations to converge. Thus we perform measurement update in several steps each time linearizing measurement function $F(\cdot)$ around the updated value $\hat{x}_n^i$. We use the equations of Iterated Extended Kalman Filter (IEKF) presented in [2]. The measurement update equations respectively for EKF and IEKF are presented in Table 5. The time update equations remain the same as before.

D. Equivalence of IEIF and Optimization 2

To show that the optimizing the objective 2 with LM algorithm is equivalent to IEKF, we first rewrite IEKF in information form as shown in [1]. The two algorithms presented in Table 6 yield the same result.

Table 5. EKF vs IEKF measurement update

Extended Kalman Filter	Iterated Extended Kalman Filter
	for $i = 1 ...$
$F_n = F(\hat{x}_n^0)$	$F_n^i = F(\hat{x}_n^i)$
$J_{n\,[u,v]} = \frac{\partial F_{[u]}}{\partial x_{[v]}}(\hat{x}_n^0)$	$J_{n\,[u,v]}^{\,i} = \frac{\partial F_{[u]}^i}{\partial x_{[v]}}(\hat{x}_n^i)$
$K_n = P_n^0 J_n^T (J_n P_n^0 J_n^T + R)^{-1}$	$K_n^i = P_n^0 {J_n^i}^T (J_n^i P_n^0 {J_n^i}^T + R)^{-1}$
$\hat{x}_n = \hat{x}_n^0 + K_n(z_n - F_n)$	$\hat{x}_n^{i+1} = \hat{x}_n^0 + K_n^i(z_n - F_n^i -$ $- J_n^i(\hat{x}_n^0 - \hat{x}_n^i))$
	end
	$\hat{x}_n = \hat{x}_n^i$
$P_n = (I - K_n J_n) P_n^0$	$P_n = (I - K_n^i J_n^i) P_n^0$

Table 6. EKF vs EIF measurement update

Extended Kalman Filter	Extended Information Filter
	$H_n = (P_n)^{-1}$
$K_n = P_n^0 J_n^T (J_n P_n^0 J_n^T + R)^{-1}$	$H_n = H_n^0 + J_n^T R^{-1} J_n$
$\hat{x}_n = \hat{x}_n^0 + K_n(z_n - F_n)$	$K_n = {H_n}^{-1} J_n^T R^{-1}$
$P_n = (I - K_n J_n) P_n^0$	$\hat{x}_n = \hat{x}_n^0 + K_n(z_n - F_n)$

Table 7 presents the measurement update for Iterated Extended Information Filter, which is equivalent to Iterated Extended Kalman Filter. We assume that measurement noise is i.i.d. for each sensor sample, thus $R = \mathrm{diag}(r)$ with r a positive scalar.

Table 7. Iterated Extended Information Filter

for $i = 1 ...$
$\quad H_n^i = \frac{1}{r}(r H_n^0 + {J_n^i}^T J_n^i)$
$\quad K_n^i = \frac{r}{r} {H_n^i}^{-1} {J_n^i}^T$
$\quad \hat{x}_n^{i+1} = \hat{x}_n^0 + K_n^i(z_n - F_n^i - J_n^i(\hat{x}_n^0 - \hat{x}_n^i))$
end
$\hat{x}_n = \hat{x}_n^i$

The LM update in Optimization **??** has the following form:

$$x_n^{i+1} = x_n^i - (\bar{J}_n^T \bar{J}_n)^{-1} \bar{J}_n^T \bar{F}_n \tag{19}$$

with $\bar{J}_n$ and $\bar{F}_n$ expanded below, $F_n^i = F(x_n^i)$, $J_n^i = \frac{\partial F}{\partial x_n}(x_n^i)$, the measurement $z_n = d_n$ and $\hat{\Sigma}_n^{-1} = r H_{n-1}$.

$$
\begin{aligned}
x_n^{i+1} = x_n^i - \left(\left[\begin{array}{c} -J_n^i \\ (rH_{n-1})^{1/2} \end{array} \right]^T \left[\begin{array}{c} -J_n^i \\ (rH_{n-1})^{1/2} \end{array} \right] \right)^{-1} \times \\
\times \left[\begin{array}{c} -J_n^i \\ (rH_{n-1})^{1/2} \end{array} \right]^T \left[\begin{array}{c} z_n - F_n^i \\ (rH_{n-1})^{1/2} (x_n^i - \hat{x}_{n-1}) \end{array} \right] = \\
= x_n^i - \underbrace{\left({J_n^i}^T J_n^i + rH_{n-1} \right)^{-1}}_{A} \times \\
\times \left(-{J_n^i}^T (z_n - F_n^i) + rH_{n-1}(x_n^i - \hat{x}_{n-1}) \right) = \\
= x_n^i + \underbrace{A {J_n^i}^T (z_n - F_n^i)}_{B} - ArH_{n-1}(x_n^i - \hat{x}_{n-1}) = \\
= \hat{x}_{n-1} + B - ArH_{n-1}(x_n^i - \hat{x}_{n-1}) + AA^{-1}(x_n^i - \hat{x}_{n-1}) = \\
= \hat{x}_{n-1} + B + A(-rH_{n-1} + {J_n^i}^T J_n^i + rH_{n-1})(x_n^i - \hat{x}_{n-1}) = \\
= \hat{x}_{n-1} + B + A{J_n^i}^T J_n^i (x_n^i - \hat{x}_{n-1}) = \\
= \hat{x}_{n-1} + \underbrace{\left({J_n^i}^T J_n^i + rH_{n-1} \right)^{-1} {J_n^i}^T}_{K_n^i} \times \\
\times (z_n - F_n^i - J_n^i(\hat{x}_{n-1} - x_n^i)),
\end{aligned}
$$

which is equivalent to the measurement update of IEIF from Table 7, with $H_n^0 = H_{n-1}$ and $x_n^0 = \hat{x}_{n-1}$.

References

[1] B. D. Anderson and J. B. Moore. Optimal filtering. *Englewood Cliffs*, 21:22–95, 1979. 5

[2] J. Havlík and O. Straka. Performance evaluation of iterated extended kalman filter with variable step-length. In *Journal of Physics: Conference Series*, volume 659, page 012022. IOP Publishing, 2015. 5

[3] D. Joseph Tan, T. Cashman, J. Taylor, A. Fitzgibbon, D. Tarlow, S. Khamis, S. Izadi, and J. Shotton. Fits like a glove: Rapid and reliable hand shape personalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5610–5619, 2016. 1

[4] P. S. Maybeck. *Stochastic models, estimation, and control*, volume 1. Academic press, 1979. 3

[5] K. B. Petersen, M. S. Pedersen, et al. The matrix cookbook. *Technical University of Denmark*, 7:15, 2008. 3

[6] A. Tkach, M. Pauly, and A. Tagliasacchi. Sphere-meshes for real-time hand modeling and tracking. *ACM Transactions on Graphics (TOG)*, 35(6):222, 2016. 1, 2

[7] G. Welch and G. Bishop. An introduction to the kalman filter. 1995. 4, 5