

Research Notes and Jacobians: Real-Time 3D Reconstruction and 6-DoF Tracking with an Event Camera

Patrick Geneva and Kevin Eickenhoff

January 5, 2018

1 Problem Statement

Real-Time 3D Reconstruction and 6-DoF with an Event Camera by Hanme Kim et al. [1] details the first example of depth estimation of features and 6 dof pose estimation using only an event camera. They present three decoupled probabilistic filters that provide 6 dof camera motion, log intensity gradient, and inverse depth estimation.

2 Ray/Triangle Intersection

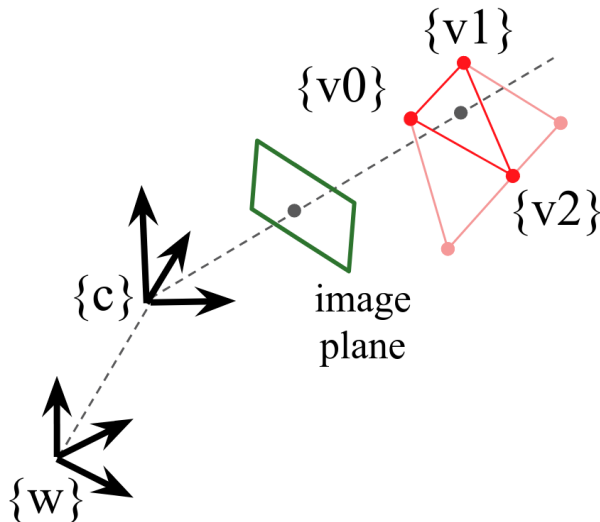


Figure 1: Pictorial view of the camera in the world frame of reference. A ray passing through the event (u, v) coordinate intersects a triangle in the 3D map of the environment in the keyframe.

Based on *Fast, Minimum Storage Ray/Triangle Intersection* by Moller et al. [2] we have the following equation that equates the intersection of the event ray with the map mesh to the interpolation between the three vertices with barycentric coordinates.

$${}^w\mathbf{p}_c + l {}^w\mathbf{R} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = (1 - a - b) {}^w\mathbf{p}_{v0} + a {}^w\mathbf{p}_{v1} + b {}^w\mathbf{p}_{v2} \quad (1)$$

where $(l, a, b)^\top$ are the barycentric coordinates in the map mesh plane. It is important to note that the (u, v) do not have any noise as they are not variables and are constant for the current

event from the sensor. We also note that the vertices ($\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2$) are points that have depth in our keyframe and thus are the map we have a known depth of. We assume that these values are “true” and thus do not have any error with them. Now using this above equality we can rearrange for the unknown quantities $(l, a, b)^\top$:

$${}^w\mathbf{p}_c + l {}^w\mathbf{R} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = (1 - a - b){}^w\mathbf{p}_{v0} + a{}^w\mathbf{p}_{v1} + b{}^w\mathbf{p}_{v2} \quad (2)$$

$$= {}^w\mathbf{p}_{v0} + ({}^w\mathbf{p}_{v1} - {}^w\mathbf{p}_{v0})a + ({}^w\mathbf{p}_{v2} - {}^w\mathbf{p}_{v0})b \quad (3)$$

$${}^w\mathbf{p}_c - {}^w\mathbf{p}_{v0} = -l {}^w\mathbf{R} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} + ({}^w\mathbf{p}_{v1} - {}^w\mathbf{p}_{v0})a + ({}^w\mathbf{p}_{v2} - {}^w\mathbf{p}_{v0})b \quad (4)$$

$${}^w\mathbf{p}_c - {}^w\mathbf{p}_{v0} = \begin{bmatrix} -{}^w\mathbf{R} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} & ({}^w\mathbf{p}_{v1} - {}^w\mathbf{p}_{v0}) & ({}^w\mathbf{p}_{v2} - {}^w\mathbf{p}_{v0}) \end{bmatrix} \begin{bmatrix} l \\ a \\ b \end{bmatrix} \quad (5)$$

Finally we get the following:

$$\begin{bmatrix} l \\ a \\ b \end{bmatrix} = \begin{bmatrix} -{}^w\mathbf{R} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} & ({}^w\mathbf{p}_{v1} - {}^w\mathbf{p}_{v0}) & ({}^w\mathbf{p}_{v2} - {}^w\mathbf{p}_{v0}) \end{bmatrix}^{-1} ({}^w\mathbf{p}_c - {}^w\mathbf{p}_{v0}) \quad (6)$$

One can note that l is the depth of the feature in the camera frame, while a and b are interpolation variables can now be used to interpolate between the three vertices’ log intensities or inverse depth values.

3 3D Pose - Kalman Filter State

We have a state that estimates the current position of the camera given all events up to time t . This state is the position of the camera pose c seen from the world frame w of reference. We denote it and its covariance as follows:

$$\mathbf{T}_{wc} = {}^w\mathbf{T}_c = \begin{bmatrix} {}^w\mathbf{R} & {}^w\mathbf{t}_c \\ \mathbf{0} & 1 \end{bmatrix} \quad (7)$$

4 3D Pose - Kalman Filter Propagation

In the paper they use a “constant position motion model”. Meaning that between event updates the camera is assumed to follow a random walk (expectation of the camera’s motion is zero). This is a reasonable assumption if the events are of high frequency. The propagation equations are defined as follows:

$$\mathbf{x}_{t|t-\tau} = \mathbf{x}_{t-\tau|t-\tau} + \mathbf{n} \quad (8)$$

$$\mathbf{P}_{t|t-\tau} = \mathbf{P}_{t-\tau|t-\tau} + \mathbf{Q} \quad (9)$$

The system is propagated from the last measurement time τ to the current time t .

5 3D Pose - Kalman Filter Update

5.1 Measurement Model

An event camera gives us events of log intensity changes of a set threshold. An event is denoted as the following:

$$\mathbf{e}(u, v) = (u, v, p, t)^\top \quad (10)$$

where (u, v) are the camera pixel location, p is the polarity (decrease or increase in log intensity), and t is a microsecond-precise timestamp. We can then define a measurement equation that maps our state into our measurement.

$$z_m = \pm C = z_k + n \quad (11)$$

$$\text{where } n \sim \mathcal{N}(0, \sigma_x^2) \quad (12)$$

From here we can define how our measurement is a function of our state as follows:

$$z_m = h(\mathbf{x}_k) + n_k \quad (13)$$

$$= z_k + n_k \quad (14)$$

$$= \mathbf{I}_L(\text{barycords}(\mathbf{x}_t)) - \mathbf{I}_L(\text{barycords}(\mathbf{x}_{t-\tau_c})) + n_k \quad (15)$$

where we define the following equations:

$$\mathbf{I}_L(\text{barycords}(\mathbf{x}_t)) = \mathbf{I}_L([l \ a \ b]^\top) \quad (16)$$

$$= (1 - a - b)I_{key}(\mathbf{v}_0) + aI_{key}(\mathbf{v}_1) + bI_{key}(\mathbf{v}_2) \quad (17)$$

$$\text{barycords}(\mathbf{x}_t) = [l \ a \ b]^\top \quad (18)$$

$$= \text{ray/triangle intersection algorithm result} \quad (19)$$

where we have a function $\mathbf{I}_L(\cdot)$ that takes in the barycentric coordinates of our current event pixel location intersected with three surrounding vertices $(\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2)$ in our keyframe depth map. It produces a log intensity value that is the interpolation between the three vertices' log intensity values. The function $\text{barycords}(\cdot)$ takes in the current state and from that computes the barycentric coordinates. This is based on the intersection of a ray passing through the event coordinates (u, v) and intersecting the plane created by three surrounding vertices. The function $I_{key}(\cdot)$ looks up the intensity value at the specified vertex seen in the keyframe image plane.

5.2 State Update Method

$$\mathbf{z}_m = h(\mathbf{x}_k) + n_k \quad (20)$$

$$= \mathbf{I}_L(\text{barycords}(\mathbf{x}_t)) - \mathbf{I}_L(\text{barycords}(\mathbf{x}_{t-\tau_c})) + n_k \quad (21)$$

It is important to note that the authors in [1] must assume that the state at the time this event pixel last received an event $\mathbf{x}_{t-\tau_c}$ is true since the state only contains the current pose \mathbf{x}_t .

$$\approx \mathbf{I}_L(\text{barycords}(\hat{\mathbf{x}}_t)) + \mathbf{H}_{il}\mathbf{H}_{bc}(\mathbf{x}_t - \hat{\mathbf{x}}_t) - \mathbf{I}_L(\text{barycords}(\hat{\mathbf{x}}_{t-\tau_c})) + \mathbf{n}_k \quad (22)$$

where we applied the chainrule on the compound function to get two Jacobians that need to be calculated.

5.3 Update Jacobian \mathbf{H}_{il}

We need to take the derivative of the equation (17) in respect to the barycentric coordinates. We can see that we are only a function of the a and b variables calculated by the $\text{barycoords}(\cdot)$ function.

$$\mathbf{H}_{il} = \left[\frac{\partial I_L}{\partial t} \quad \frac{\partial I_L}{\partial a} \quad \frac{\partial I_L}{\partial b} \right]_{1 \times 3} \quad (23)$$

$$= \left[0 \quad \frac{\partial I_L}{\partial a} \quad \frac{\partial I_L}{\partial b} \right]_{1 \times 3} \quad (24)$$

It is easy to see that the derivative of equation (17) in respect to a and b are the following:

$$\boxed{\mathbf{H}_{il} = \left[0 \quad \left(I_{key}(\mathbf{v}_1) - I_{key}(\mathbf{v}_0) \right) \quad \left(I_{key}(\mathbf{v}_2) - I_{key}(\mathbf{v}_0) \right) \right]_{1 \times 3}} \quad (25)$$

5.4 Update Jacobian \mathbf{H}_{bc}

We need to take the derivative of the equation (6) in respect to the state. This involves an inverse, thus special care needs to be taken when performing the derivative.

$$\mathbf{H}_{bc} = \frac{\partial}{\partial \hat{\mathbf{x}}_t} (\mathbf{M}^{-1} \mathbf{D}) \quad (26)$$

where \mathbf{M} is the first matrix in equation (6) and \mathbf{D} is the second part in equation (6). Using the product rule and Section 2.2 Derivatives of an inverse from the Matrix Cookbook [3] we can say that our Jacobian is the following:

$$= - \frac{\partial \mathbf{M}^{-1}}{\partial \hat{\mathbf{x}}_t} \mathbf{D} + \mathbf{M}^{-1} \frac{\partial \mathbf{D}}{\partial \hat{\mathbf{x}}_t} \quad (27)$$

$$= \text{composition} \left(- \mathbf{M}^{-1} \frac{\partial \mathbf{M}}{\partial \hat{x}(i)_t} \mathbf{M}^{-1} \mathbf{D} + \mathbf{M}^{-1} \frac{\partial \mathbf{D}}{\partial \hat{x}(i)_t} \right) \quad (28)$$

where the ‘‘composition’’ is the element-wise derivative of both \mathbf{M} and \mathbf{D} . This element-wise derivative is required as the inverse identity from the Matrix Cookbook [3] is only valid when taking the derivative in respect to a scalar. We can calculate those as follows:

$$\frac{\partial \mathbf{M}}{\partial \hat{\theta}(i)_t} = \begin{bmatrix} w \mathbf{R} & \left[\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \times \right] \mathbf{e}_i & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \end{bmatrix}_{3 \times 3} \quad (29)$$

$$\frac{\partial \mathbf{M}}{\partial \hat{p}(i)_t} = \mathbf{0}_{3 \times 3} \quad (30)$$

$$\frac{\partial \mathbf{D}}{\partial \hat{\theta}(i)_t} = \mathbf{0}_{3 \times 1} \quad (31)$$

$$\frac{\partial \mathbf{D}}{\partial \hat{p}(i)_t} = \mathbf{e}_i \quad (32)$$

where we define \mathbf{e}_i as the unit vector in the direction of that measurement (e.g., $\mathbf{e}_x = [1 \ 0 \ 0]^\top$ for the x-direction derivative). By composing them, we can get the following large Jacobian matrix:

$$\mathbf{H}_{bc} = \left[\begin{array}{ccc} \frac{\partial}{\partial \hat{\theta}(x)_t} (\mathbf{M}^{-1} \mathbf{D}) & \frac{\partial}{\partial \hat{\theta}(y)_t} (\mathbf{M}^{-1} \mathbf{D}) & \frac{\partial}{\partial \hat{\theta}(z)_t} (\mathbf{M}^{-1} \mathbf{D}) \\ \frac{\partial}{\partial \hat{p}(x)_t} (\mathbf{M}^{-1} \mathbf{D}) & \frac{\partial}{\partial \hat{p}(y)_t} (\mathbf{M}^{-1} \mathbf{D}) & \frac{\partial}{\partial \hat{p}(z)_t} (\mathbf{M}^{-1} \mathbf{D}) \end{array} \right]_{3 \times 6} \quad (33)$$

$$= [\mathbf{J}_1 \ \mathbf{J}_2 \ \mathbf{J}_3 \ \mathbf{J}_4 \ \mathbf{J}_5 \ \mathbf{J}_6]_{3 \times 6} \quad (34)$$

We the following final Jacobian values:

$$\mathbf{J}_1 = -\mathbf{M}^{-1} \left[\begin{array}{ccc} w_c \mathbf{R} \left[\begin{array}{c} u \\ v \\ 1 \end{array} \right] \times \mathbf{e}_x & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \end{array} \right]_{3 \times 3} \mathbf{M}^{-1} \mathbf{D} + \mathbf{M}^{-1} \mathbf{0}_{3 \times 1} \quad (35)$$

$$= -\mathbf{M}^{-1} \left[\begin{array}{ccc} w_c \mathbf{R} \left[\begin{array}{c} u \\ v \\ 1 \end{array} \right] \times \mathbf{e}_x & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \end{array} \right]_{3 \times 3} \mathbf{M}^{-1} \mathbf{D} \quad (36)$$

$$\mathbf{J}_2 = -\mathbf{M}^{-1} \left[\begin{array}{ccc} w_c \mathbf{R} \left[\begin{array}{c} u \\ v \\ 1 \end{array} \right] \times \mathbf{e}_y & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \end{array} \right]_{3 \times 3} \mathbf{M}^{-1} \mathbf{D} + \mathbf{M}^{-1} \mathbf{0}_{3 \times 1} \quad (37)$$

$$= -\mathbf{M}^{-1} \left[\begin{array}{ccc} w_c \mathbf{R} \left[\begin{array}{c} u \\ v \\ 1 \end{array} \right] \times \mathbf{e}_y & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \end{array} \right]_{3 \times 3} \mathbf{M}^{-1} \mathbf{D} \quad (38)$$

$$\mathbf{J}_3 = -\mathbf{M}^{-1} \left[\begin{array}{ccc} w_c \mathbf{R} \left[\begin{array}{c} u \\ v \\ 1 \end{array} \right] \times \mathbf{e}_z & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \end{array} \right]_{3 \times 3} \mathbf{M}^{-1} \mathbf{D} + \mathbf{M}^{-1} \mathbf{0}_{3 \times 1} \quad (39)$$

$$= -\mathbf{M}^{-1} \left[\begin{array}{ccc} w_c \mathbf{R} \left[\begin{array}{c} u \\ v \\ 1 \end{array} \right] \times \mathbf{e}_z & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \end{array} \right]_{3 \times 3} \mathbf{M}^{-1} \mathbf{D} \quad (40)$$

$$\mathbf{J}_4 = -\mathbf{M}^{-1} \mathbf{0}_{3 \times 3} \mathbf{M}^{-1} \mathbf{D} + \mathbf{M}^{-1} \mathbf{e}_x \quad (41)$$

$$= \mathbf{M}^{-1} \mathbf{e}_x \quad (42)$$

$$\mathbf{J}_5 = -\mathbf{M}^{-1} \mathbf{0}_{3 \times 3} \mathbf{M}^{-1} \mathbf{D} + \mathbf{M}^{-1} \mathbf{e}_y \quad (43)$$

$$= \mathbf{M}^{-1} \mathbf{e}_y \quad (44)$$

$$\mathbf{J}_6 = -\mathbf{M}^{-1} \mathbf{0}_{3 \times 3} \mathbf{M}^{-1} \mathbf{D} + \mathbf{M}^{-1} \mathbf{e}_z \quad (45)$$

$$= \mathbf{M}^{-1} \mathbf{e}_z \quad (46)$$

6 Pixel-Wise Inverse Depth - Kalman Filter Update

6.1 Measurement Model

For a given pixel from an event, we want to update our inverse depth map of the current keyframe. We can define the inverse depth of the vertices of the plan in the depth map that the pixel from

the event intersects as the following:

$$\boldsymbol{\rho} = (\rho_{v0}, \rho_{v1}, \rho_{v2})^\top \quad (47)$$

where $\boldsymbol{\rho}$ is the stacked inverse depths of the specified . We can then define a measurement equation that maps our state into our measurement.

$$z_m = \pm C = z_k + n \quad (48)$$

$$\text{where } n \sim \mathcal{N}(0, \sigma_p^2) \quad (49)$$

From here we can define how our measurement is a function of our state as follows:

$$z_m = h(\boldsymbol{\rho}_t) + n_k \quad (50)$$

$$= z_k + n_k \quad (51)$$

$$= I_L(\text{barycords}(\boldsymbol{\rho}_t)) - I_L(\text{barycords}(\boldsymbol{\rho}_{t-\tau_c})) + n_k \quad (52)$$

where we define the following equations:

$$I_L(\text{barycords}(\boldsymbol{\rho}_t)) = I_L([l \ a \ b]^\top) \quad (53)$$

$$= (1 - a - b)I_{key}(\mathbf{v}_0) + aI_{key}(\mathbf{v}_1) + bI_{key}(\mathbf{v}_2) \quad (54)$$

$$\text{barycords}(\boldsymbol{\rho}_t) = [l \ a \ b]^\top \quad (55)$$

$$= \text{ray/triangle intersection algorithm result} \quad (56)$$

$$= \left[-{}^w\mathbf{R}_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad \left(\text{inv}(\rho_{v1}) - \text{inv}(\rho_{v0}) \right) \quad \left(\text{inv}(\rho_{v2}) - \text{inv}(\rho_{v0}) \right) \right]^{-1} \left({}^w\mathbf{p}_c - \text{inv}(\rho_{v0}) \right)$$

$$\text{inv}(\rho_{vi}) = \frac{{}^w\mathbf{R}_{key}}{\rho_{vi}} \begin{bmatrix} u_{vi} \\ v_{vi} \\ 1 \end{bmatrix} + {}^w\mathbf{p}_{key} \quad (57)$$

where we have a function $I_L(\cdot)$ that takes in barycentric coordinates of our current event pixel location intersected with three surrounding vertices ($\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2$) in our keyframe depth map. It produces a log intensity value that is the interpolation between the three vertices' log intensity values.

Different from the 3D pose estimation, we use inverse depth parameterization here with the ρ values that we are looking to update. It should be noted that since the authors model each pixel as independent, it is impossible to update both $\boldsymbol{\rho}_t$ and $\boldsymbol{\rho}_{t-\tau_c}$ in the same filter. To solve this, we can update both independently using the same measurement using independent filters.

6.2 State Update Method

$$\mathbf{z}_m = h(\boldsymbol{\rho}_t) + n_k \quad (58)$$

$$= I_L(\text{barycords}(\boldsymbol{\rho}_t)) - I_L(\text{barycords}(\boldsymbol{\rho}_{t-\tau_c})) + n_k \quad (59)$$

It is important to note that the authors in [1] must assume that the $\boldsymbol{\rho}_{t-\tau_c}$ is true since the only thing that is estimated is the three current camera vertices $\boldsymbol{\rho}_t$.

$$\approx \mathbf{I}_L(\text{barycords}(\hat{\boldsymbol{\rho}}_t)) + \mathbf{H}_{il}\mathbf{H}_{bci}(\boldsymbol{\rho}_t - \hat{\boldsymbol{\rho}}_t) - \mathbf{I}_L(\text{barycords}(\hat{\boldsymbol{\rho}}_{t-\tau_c})) + \mathbf{n}_k \quad (60)$$

where we applied the chainrule on the compound function to get two Jacobians that need to be calculated.

6.3 Update Jacobian \mathbf{H}_{il}

We need to take the derivative of the equation (54) in respect to the barycentric coordinates. We can see that we are only a function of the a and b variables calculated by the $\text{barycords}(\cdot)$ function.

$$\mathbf{H}_{il} = \left[\frac{\partial \mathbf{I}_L}{\partial l} \quad \frac{\partial \mathbf{I}_L}{\partial a} \quad \frac{\partial \mathbf{I}_L}{\partial b} \right]_{1 \times 3} \quad (61)$$

$$= \left[0 \quad \frac{\partial \mathbf{I}_L}{\partial a} \quad \frac{\partial \mathbf{I}_L}{\partial b} \right]_{1 \times 3} \quad (62)$$

It is easy to see that the derivative of equation (54) in respect to a and b are the following:

$$\boxed{\mathbf{H}_{il} = \left[0 \quad \left(I_{key}(\mathbf{v}_1) - I_{key}(\mathbf{v}_0) \right) \quad \left(I_{key}(\mathbf{v}_2) - I_{key}(\mathbf{v}_0) \right) \right]_{1 \times 3}} \quad (63)$$

6.4 Inverse Feature Jacobian \mathbf{H}_{inv}

Original measurement equation:

$$\text{inv}(\rho_{vi}) = \frac{w}{\rho_{vi}} \mathbf{R} \begin{bmatrix} u_{vi} \\ v_{vi} \\ 1 \end{bmatrix} + {}^w \mathbf{p}_{key} \quad (64)$$

Perturbing each side of the equation we can do the following:

$${}^c \hat{\mathbf{p}}_f + {}^c \tilde{\mathbf{p}}_f = \frac{w}{\hat{\rho}_{vi} + \tilde{\rho}_{vi}} \mathbf{R} \begin{bmatrix} u_{vi} \\ v_{vi} \\ 1 \end{bmatrix} + {}^w \mathbf{p}_{key} \quad (65)$$

Adding zero to the left side in the form of $\frac{w}{\hat{\rho}_{vi}} \mathbf{R} \begin{bmatrix} u_{vi} & v_{vi} & 1 \end{bmatrix}^\top - \frac{w}{\hat{\rho}_{vi}} \mathbf{R} \begin{bmatrix} u_{vi} & v_{vi} & 1 \end{bmatrix}^\top$ we have the following:

$${}^c \hat{\mathbf{p}}_f + {}^c \tilde{\mathbf{p}}_f = \frac{w}{\hat{\rho}_{vi} + \tilde{\rho}_{vi}} \mathbf{R} \begin{bmatrix} u_{vi} \\ v_{vi} \\ 1 \end{bmatrix} + {}^w \mathbf{p}_{key} + \frac{w}{\hat{\rho}_{vi}} \mathbf{R} \begin{bmatrix} u_{vi} \\ v_{vi} \\ 1 \end{bmatrix} - \frac{w}{\hat{\rho}_{vi}} \mathbf{R} \begin{bmatrix} u_{vi} \\ v_{vi} \\ 1 \end{bmatrix} \quad (66)$$

$$= {}^c \hat{\mathbf{p}}_f + \frac{w}{\hat{\rho}_{vi} + \tilde{\rho}_{vi}} \mathbf{R} \begin{bmatrix} u_{vi} \\ v_{vi} \\ 1 \end{bmatrix} - \frac{w}{\hat{\rho}_{vi}} \mathbf{R} \begin{bmatrix} u_{vi} \\ v_{vi} \\ 1 \end{bmatrix} \quad (67)$$

$$= {}^c \hat{\mathbf{p}}_f + \frac{w}{\hat{\rho}_{vi}} \mathbf{R} \frac{\hat{\rho}_{vi} - (\hat{\rho}_{vi} + \tilde{\rho}_{vi})}{(\hat{\rho}_{vi} + \tilde{\rho}_{vi})\hat{\rho}_{vi}} \begin{bmatrix} u_{vi} \\ v_{vi} \\ 1 \end{bmatrix} \quad (68)$$

$$= {}^c \hat{\mathbf{p}}_f - \frac{w}{\hat{\rho}_{vi}} \mathbf{R} \frac{\tilde{\rho}_{vi}}{\hat{\rho}_{vi}^2 + \tilde{\rho}_{vi}\hat{\rho}_{vi}} \begin{bmatrix} u_{vi} \\ v_{vi} \\ 1 \end{bmatrix} \quad (69)$$

Multiplying by the conjugate of the denominator $\hat{\rho}_{vi}^2 - \tilde{\rho}_{vi}\hat{\rho}_{vi}$ we have the following:

$$= {}^c\hat{\mathbf{P}}_f - {}^w_{key}\mathbf{R} \frac{\tilde{\rho}_{vi}(\hat{\rho}_{vi}^2 - \tilde{\rho}_{vi}\hat{\rho}_{vi})}{\hat{\rho}_{vi}^4 - \tilde{\rho}_{vi}^2\hat{\rho}_{vi}^2} \begin{bmatrix} u_{vi} \\ v_{vi} \\ 1 \end{bmatrix} \quad (70)$$

Removing the negligible squared error terms we have:

$$= {}^c\hat{\mathbf{P}}_f - {}^w_{key}\mathbf{R} \frac{\tilde{\rho}_{vi}\hat{\rho}_{vi}^2}{\hat{\rho}_{vi}^4} \begin{bmatrix} u_{vi} \\ v_{vi} \\ 1 \end{bmatrix} \quad (71)$$

$$= {}^c\hat{\mathbf{P}}_f - {}^w_{key}\mathbf{R} \frac{\tilde{\rho}_{vi}}{\hat{\rho}_{vi}^2} \begin{bmatrix} u_{vi} \\ v_{vi} \\ 1 \end{bmatrix} \quad (72)$$

$$= {}^c\hat{\mathbf{P}}_f + \left(- {}^w_{key}\mathbf{R} \frac{1}{\hat{\rho}_{vi}^2} \begin{bmatrix} u_{vi} \\ v_{vi} \\ 1 \end{bmatrix} \right) \tilde{\rho}_{vi} \quad (73)$$

Thus we have the following jacobian:

$$\boxed{\mathbf{H}_{inv} = - {}^w_{key}\mathbf{R} \frac{1}{\hat{\rho}_{vi}^2} \begin{bmatrix} u_{vi} \\ v_{vi} \\ 1 \end{bmatrix}} \quad (74)$$

6.5 Update Jacobian \mathbf{H}_{bci}

We need to take the derivative of the equation (57) in respect to the inverse depth values. This involves an inverse, thus special care needs to be taken when performing the derivative.

$$\mathbf{H}_{bc} = \frac{\partial}{\partial \hat{\rho}_t} (\mathbf{M}^{-1}\mathbf{D}) \quad (75)$$

where \mathbf{M} is the first matrix in equation (57) and \mathbf{D} is the second part in equation (57). Using the product rule and Section 2.2 Derivatives of an inverse from the Matrix Cookbook [3] we can say that our Jacobian is the following:

$$= - \frac{\partial \mathbf{M}^{-1}}{\partial \hat{\rho}_t} \mathbf{D} + \mathbf{M}^{-1} \frac{\partial \mathbf{D}}{\partial \hat{\rho}_t} \quad (76)$$

$$= \text{composition} \left(- \mathbf{M}^{-1} \frac{\partial \mathbf{M}}{\partial \hat{\rho}(i)_t} \mathbf{M}^{-1} \mathbf{D} + \mathbf{M}^{-1} \frac{\partial \mathbf{D}}{\partial \hat{\rho}(i)_t} \right) \quad (77)$$

where the “composition” is the element-wise derivative of both \mathbf{M} and \mathbf{D} . This element-wise derivative is required as the inverse identity from the Matrix Cookbook [3] is only valid when

taking the derivative in respect to a scalar. We can calculate those as follows:

$$\frac{\partial \mathbf{M}}{\partial \hat{\rho}(v0)_t} = [\mathbf{0}_{3 \times 1} \quad -\mathbf{H}_{inv} \quad -\mathbf{H}_{inv}]_{3 \times 3} \quad (78)$$

$$\frac{\partial \mathbf{M}}{\partial \hat{\rho}(v1)_t} = [\mathbf{0}_{3 \times 1} \quad \mathbf{H}_{inv} \quad \mathbf{0}_{3 \times 1}]_{3 \times 3} \quad (79)$$

$$\frac{\partial \mathbf{M}}{\partial \hat{\rho}(v2)_t} = [\mathbf{0}_{3 \times 1} \quad \mathbf{0}_{3 \times 1} \quad \mathbf{H}_{inv}]_{3 \times 3} \quad (80)$$

$$\frac{\partial \mathbf{D}}{\partial \hat{\rho}(v0)_t} = -\mathbf{H}_{inv} \quad (81)$$

$$\frac{\partial \mathbf{D}}{\partial \hat{\rho}(v1)_t} = \mathbf{0}_{3 \times 1} \quad (82)$$

$$\frac{\partial \mathbf{D}}{\partial \hat{\rho}(v2)_t} = \mathbf{0}_{3 \times 1} \quad (83)$$

By composing them, we can get the following large Jacobian matrix:

$$\mathbf{H}_{bc} = \left[\begin{array}{ccc} \frac{\partial}{\partial \hat{\rho}(v0)_t} (\mathbf{M}^{-1} \mathbf{D}) & \frac{\partial}{\partial \hat{\rho}(v1)_t} (\mathbf{M}^{-1} \mathbf{D}) & \frac{\partial}{\partial \hat{\rho}(v2)_t} (\mathbf{M}^{-1} \mathbf{D}) \end{array} \right]_{3 \times 3} \quad (84)$$

$$= [\mathbf{J}_1 \quad \mathbf{J}_2 \quad \mathbf{J}_3]_{3 \times 3} \quad (85)$$

We the following final Jacobian values:

$$\boxed{\mathbf{J}_1 = -\mathbf{M}^{-1} \left[\mathbf{0}_{3 \times 1} \quad -\mathbf{H}_{inv} \quad -\mathbf{H}_{inv} \right]_{3 \times 3} \mathbf{M}^{-1} \mathbf{D} - \mathbf{M}^{-1} \mathbf{H}_{inv}} \quad (86)$$

$$\boxed{\mathbf{J}_2 = -\mathbf{M}^{-1} \left[\mathbf{0}_{3 \times 1} \quad \mathbf{H}_{inv} \quad \mathbf{0}_{3 \times 1} \right]_{3 \times 3} \mathbf{M}^{-1} \mathbf{D}} \quad (87)$$

$$\boxed{\mathbf{J}_3 = -\mathbf{M}^{-1} \left[\mathbf{0}_{3 \times 1} \quad \mathbf{0}_{3 \times 1} \quad \mathbf{H}_{inv} \right]_{3 \times 3} \mathbf{M}^{-1} \mathbf{D}} \quad (88)$$

References

- [1] Hanme Kim, Stefan Leutenegger, and Andrew J Davison. “Real-time 3D reconstruction and 6-DoF tracking with an event camera”. In: *European Conference on Computer Vision*. Springer. 2016, pp. 349–364.
- [2] Tomas Möller and Ben Trumbore. “Fast, minimum storage ray/triangle intersection”. In: *ACM SIGGRAPH 2005 Courses*. ACM. 2005, p. 7.
- [3] Kaare Brandt Petersen, Michael Syskind Pedersen, et al. “The matrix cookbook”. In: *Technical University of Denmark 7* (2008), p. 15.