# Supplementary Material

Anonymous ICCV submission

Paper ID

## 1. Derivations of Computational Complexity

| | EKF | | SRIF | |
|---|---|---|---|---|
| | Step | Complexity | Step | Complexity |
| **Update** | | | | |
| | $\mathbf{PH}^T$ | $288N^2$ | $\mathbf{A} = \mathbf{Q}\begin{bmatrix}\mathbf{R}\\ \mathbf{0}\end{bmatrix}$ | $144N^3$ |
| | $\mathbf{S} = \mathbf{HPH}^T + \mathbf{Q}_v$ | $100N^2$ | $\mathbf{Q}^T\Delta\mathbf{e}_{k+1}$ | $120N^2$ |
| | $\mathbf{S} = \mathbf{R}_s^T\mathbf{R}_s$ | $8N^3/3$ | $\mathbf{R}^{-1}\mathbf{Q}^T\Delta\mathbf{e}_{k+1}$ | $36N^2$ |
| | $\mathbf{K} = \mathbf{PH}^T\mathbf{R}_s^{-1}\mathbf{R}_s^{-T}$ | $48N^3$ | | |
| | $\mathbf{P} = \mathbf{P} - \mathbf{KHP}$ | $144N^3$ | | |
| **Predict** | | | | |
| | | Negligible | | Negligible |
| **Feature removal** | | | | |
| | | | $\mathrm{QR}(\mathbf{R}_\Pi)$ | $324Nk(N-k)$<br>$162N^{3\dagger}$ |
| **Total** | | | | |
| | | $194.67N^3$ | | $306N^{3\dagger}$ |

Table 1. Computational complexity of EKF and SRIF, $^\dagger$Upper Bound.

Derivations for the estimates of computational complexity in Table 2 of the main paper are presented here. The table is reproduced here for convenience of the reader. Various results from the classic text of Golub and Van Loan are used [1].

- Computing $\mathbf{AB}$ where $\mathbf{A} \in \mathbb{R}^{m\times k}$ and $\mathbf{B} \in \mathbb{R}^{k\times n}$ requires $2mnk$ ops.
- The Cholesky factorization $\mathbf{A} = \mathbf{R}^\mathbf{T}\mathbf{R}$, where $\mathbf{A} \in \mathbb{R}^{n\times n}$ is symmetric positive-definite, requires $n^3/3$ ops.
- Solving $\mathbf{Rx} = \mathbf{y}$ for upper/lower triangular $\mathbf{R} \in \mathbb{R}^{n\times n}$ $\mathbf{R}$ requires $n^2$ ops.
- Computing the QR factorization $\mathbf{A} = \mathbf{QR}$, $\mathbf{A} \in \mathbb{R}^{m\times n}$ requires $2n^2(m - n/3)$ ops. Note that the $\mathbf{Q}$ matrix is never calculated explicitly but is represented by the Householder vectors obtained during factorization.

### Complexity of $\mathbf{PH}^T$

Direct multiplication of $\mathbf{P} \in \mathbb{R}^{6N\times 6N}$ and $\mathbf{H} \in \mathbb{R}^{2N\times 6N}$ will require $2 \times 2N \times 6N \times 6N = 144N^3$ ops. Note, however, that $\mathbf{H}$ is sparse and has only 12 non-zero elements per row. Hence, the number of ops reduces to $2\times 2N \times 12 \times 6N = 288N^2$.

### Complexity of $\mathbf{S} = \mathbf{HPH}^T + \mathbf{Q}_v$

$\mathbf{PH}^T$, which has already been calculated above, is of dimension $6N \times 2N$. Direct multiplication of $\mathbf{H}$ with $\mathbf{PH}^T$ would require $2 \times 2N \times 6N \times 2N = 48N^3$ ops. Again, leveraging the sparsity of $\mathbf{H}$ yields, $2 \times 2N \times 12 \times 2N = 96N^2$ ops. Addition of $\mathbf{Q}_v$ requires an additional $2N \times 2N = 4N^2$ ops bringing the total to $100N^2$ ops.

### Complexity of $\mathbf{S} = \mathbf{R}_s^T \mathbf{R}_s$

Cholesky of $\mathbf{S} \in \mathbb{R}^{2N \times 2N}$, requires $(2N)^3/3 = 8N^3/3$ ops.

### Complexity of $\mathbf{K} = \mathbf{P}\mathbf{H}^T \mathbf{R}_s^{-1} \mathbf{R}_s^{-T}$

Calculating $\mathbf{K}$ requires solving $\mathbf{P}\mathbf{H}^T$, a $6N \times 2N$ matrix, by a upper-triangular matrix, $\mathbf{R}_s \in \mathbb{R}^{2N \times 2N}$, followed by a its transposed lower-triangular version $\mathbf{R}_s^T$. Consider first, the lower-triangular solving. Solving one row of $\mathbf{P}\mathbf{H}^T$ by $\mathbf{R}_s$ requires $2N \times 2N$ ops. Hence, solving all $6N$ rows of $\mathbf{P}\mathbf{H}^T$ requires $24N^3$ ops. Similarly, the upper-triangular solving will require another $24N^3$ ops, bringing the total to $48N^3$ ops.

### Complexity of $\mathbf{P} = \mathbf{P} - \mathbf{K}\mathbf{H}\mathbf{P}$

Here, it is assumed that $\mathbf{H}\mathbf{P}$ is already available by transposting $\mathbf{P}\mathbf{H}^T$ above. Multiplication of $\mathbf{K}$ by $\mathbf{H}\mathbf{P}$ requires $2 \times 6N \times 2N \times 6N = 144N^3$. Substracting from $\mathbf{P}$ is an $O(N^2)$ operation, which is negligible.

### Complexity of $\mathbf{A} = \mathbf{Q} \begin{bmatrix} \mathbf{R} \\ 0 \end{bmatrix}$

Matrix $\mathbf{A}$ is a $8N \times 6N$ matrix and has the following form (time susbscripts dropped for notational convenience)

$$\mathbf{A} = \begin{bmatrix} \mathbf{R}^x \\ \mathbf{R}^v \mathbf{H} \end{bmatrix} = \begin{bmatrix} \times & \times & \cdots & \times \\ 0 & \times & \cdots & \times \\ 0 & 0 & \ddots & \times \\ 0 & 0 & \cdots & \times \\ \times & \times & \cdots & \times \\ \times & \times & \cdots & \times \end{bmatrix}$$

where, $\times$ denotes a non-zero element. Computing the QR-factorization of $\mathbf{A}$ would normally require approximately $4 \times 8N \times (6N)^2 - (6N)^3/3 = 1080N^3$ ops. However, the upper $6N \times 6N$ portion of $\mathbf{A}$, i.e $\mathbf{R}^x$, is already upper-triangular. In order to transform it into $\begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}$ by QR factorization, therefore, only the lower $2N \times 6N$ portion of $\mathbf{A}$ needs to be zeroed out. A sequence of $6N$ Householder transformations are applied. The $l$th transformation operates on the lower $(8N - l + 1) \times (6N - l + 1)$ sub-matrix $\mathbf{A}_l$ of $\mathbf{A}$. The Householder vector, $\mathbf{v}_l$, therefore, has length $8N - l + 1$, but it has only $2N + 1$ non-zero elements. Hence, in the computation of the Householder transformation

$$(\mathbf{I} - \beta_l \mathbf{v}_l \mathbf{v}_l^T)\mathbf{A}_l = \mathbf{A}_l - \beta_l \mathbf{v}_l (\mathbf{v}_l^T \mathbf{A}_l),$$

$2(1 + 2N)(6N - l)$ ops are required for the computation of $\mathbf{v}_l^T \mathbf{A}_l$; a further $(1 + 2N)(6N - l)$ ops are required for the computation of the outer-product $\beta_l \mathbf{v}_l (\mathbf{v}_l^T \mathbf{A}_l)$; and finally $(1 + 2N)(6N - l)$ ops are required for the computation of the difference $\mathbf{A}_l - \beta_l \mathbf{v}_l (\mathbf{v}_l^T \mathbf{A}_l)$, which totals $4(1 + 2N)(6N - l)$. Summing this, yields

$$\sum_{l=0}^{l=6N-1} 4(1 + 2N)(6N - l) \approx 2 \times 2N \times (6N)^2 = 144N^3,$$

which is a significant reduction.

### Complexity of $\mathbf{Q}^T \Delta \mathbf{e}_{k+1}$

From [1, Chapter 5], calculation of $\mathbf{Q}^T \mathbf{C}$, where $\mathbf{Q} \in \mathbb{R}^{n \times n}$ is a product of $r$ Householder matrices, and $\mathbf{C} \in \mathbb{R}^{n \times q}$, requires $2qr(2n - r)$ ops rather than $2n^2 q$ ops that would be required for direct multiplication. In this case, $r = 6N$, $n = 8N$, and $q = 1$. Hence, number of ops is $2 \times 6N \times (16N - 6N) = 120N^2$.

### Complexity of $\mathbf{R}^{-1} \mathbf{Q}^T \Delta \mathbf{e}_{k+1}$

This is simply solving of a vector, $\mathbf{Q}^T \Delta \mathbf{e}_{k+1}$, by a lower-triangular matrix $\mathbf{R}^{-1}$ which is $6N \times 6N$. This requires $36N^2$ ops.

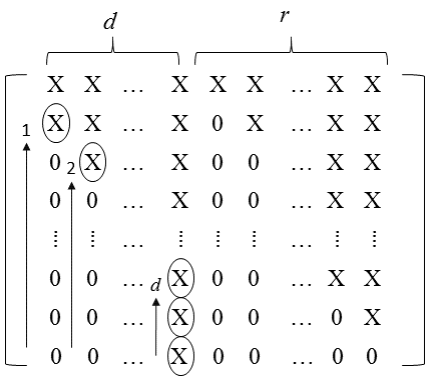


Figure 1. EKF-based Structure-from-Motion.

## Complexity of triangularizing $\mathbf{R}_\Pi$

The cost of feature-reduction step in SRIF-SfM comes from triangularizing $\mathbf{R}_\Pi$. Suppose, for simplicity, that $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{R} \in \mathbb{R}^{n \times n}$, $d$ elements from $\mathbf{x}$ are discarded and $r = n - d$ elements are retained. Recall, from the main manuscript, that $\mathbf{R}_\Pi = \mathbf{R}\mathbf{\Pi}^T$, where $\mathbf{\Pi}$ is a permutation matrix; hence $\mathbf{R}_\Pi$ is a column-permutation of an upper-triangular $\mathbf{R}$. The permutation matrix $\mathbf{\Pi}$ is such that the columns corresponding to the $d$ elements to be discarded are placed first, followed by the $r$ columns corresponding to the elements to be retained. The general structure of $\mathbf{R}_\Pi$ is shown in Figure 1. Triangularizing $\mathbf{R}_\Pi$ entails applying a sequence of Givens rotations in the order shown in the figure to zero out the elements below the principal diagonal. It is clear that greater the number of non-zero elements below the principal diagonal, more will be the number of Givens rotations required to triangularize $\mathbf{R}_\Pi$. In general, this number will depend not just on the number of elements to be discarded from $\mathbf{x}$, but on the specific order of the elements being removed. Given this dependency, it is not possible to derive a closed-form expression for the number of operations required. However, it is possible to obtain an expression for the upper bound.

The upper-bound will occur when the last $d$ elements of $\mathbf{x}$ need to be removed, in which case the permutation matrix $\mathbf{\Pi}$ would be such that the last $d$ columns of $\mathbf{R}$ are permuted to the first $d$ positions. In this case, it is required to zero out the lowest $n - d = r$ non-zero elements in the first $d$ columns. Consider the lowest non-zero element in the first column. Zeroing this out via a Givens rotation will impact the corresponding elements in the last $d$ columns. Thus the $2 \times 2$ Givens rotation would have effectively been applied to $d + 1$ columns. Next, we proceed to the element above this. Zeroing this out would impact the corresponding elements in the last $d$ columns, resulting in effectively applying the Givens rotation to $d + 2$ columns. Proceeding thus to the element just below the topmost in the first column, it can be seen that zeroing this would involve Givens rotation to $n$ columns. Thus for the 1st column, the total number of Givens rotations involved are:

$$\sum_{i=d+1}^{n} i \tag{1}$$

Proceeding similarly for the $j$th column yields the total number of Givens rotation as

$$\sum_{i=d-j+2}^{n-j+1} i \tag{2}$$

Thus, the total number of Givens rotations for the first $d$ columns is

$$\sum_{j=1}^{d} \left( \sum_{i=d-j+2}^{n-j+1} i \right) = \frac{d(n-d)(n+2)}{2} \tag{3}$$

The above expression attains a maximum at $d = n/2$ and its value is $\approx n^3/8$. Now, each Givens rotation requires 6 ops [1, chapter 5]. Hence, total number of ops is $3n^3/4$. Since $n = 6N$ in our case, the number of ops is

$$3\frac{(6N)^3}{4} = 162N^3$$

## 2. Dataset and Results

In this section we present the *pose-only* plots for ten Kitti odometry clips for which pose and position errors are given in Table 1 of the manuscript. The pose is represented by a 4-element quaternion $\mathbf{q} \triangleq (x, y, z, w)$. Thus in the following plots, the four elements of the pose-quaternion are plotted against the frame index. For the selected ten Kitti clips, the ground-truth information is available. We generated outputs for four different cases: EKF single precision, EKF 64bit fixed-point, SRIF single precision, and SRIF 32bit fixed-point implementations. The errors in the pose quaternion are a good indication of the quality of the quality of ego-motion tracking and also the estimated 3D scene structure. Since our algorithm is a monocular SfM, the scene structure and self-position values suffer from scale ambiguity. Hence the position plots do not lend themselves to an intuitive visual comparison. We have compared the ground-truth pose (solid line) values with the four implementations

1. EKF single precision (long-dash)
2. EKF 64bit fixed-point (dots and dash)
3. SRIF single precision and (dots)
4. SRIF 32bit fixed-point implementations (short dash)

It can be observed from the plots that the estimated outputs are match so closely with the ground-truth as to be largely indistinguishable from it. A slightly higher deviation in the estimated values for sequence 02 (Figure 3) results in higher position errors (Table 1 in the manuscript).

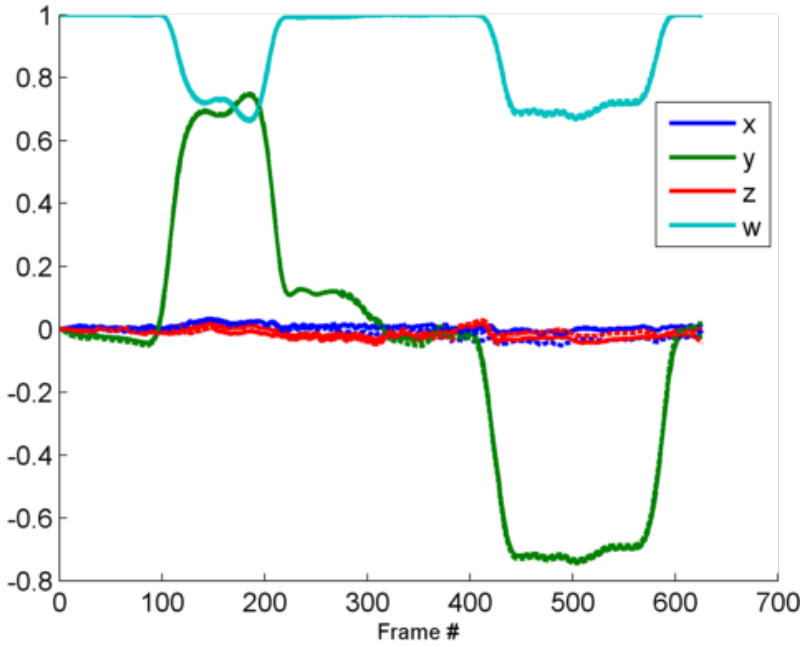


Figure 2. Pose results for kitti odometry clip 00 (Seq#1 in Table 2). The solid-lines are for the ground-truth pose values. The dotted-lines are for the estimated pose values from EKF single precision, EKF 64bit fixed-point, SRIF single precision and SRIF 32bit fixed-point implementations (in most cases overlapping with the ground-truth plots).

## References

[1] G. Golub and C. V. Loan. *Matrix Computations*. Johns Hopkins University Press, third edition, 1996. 1, 2, 3

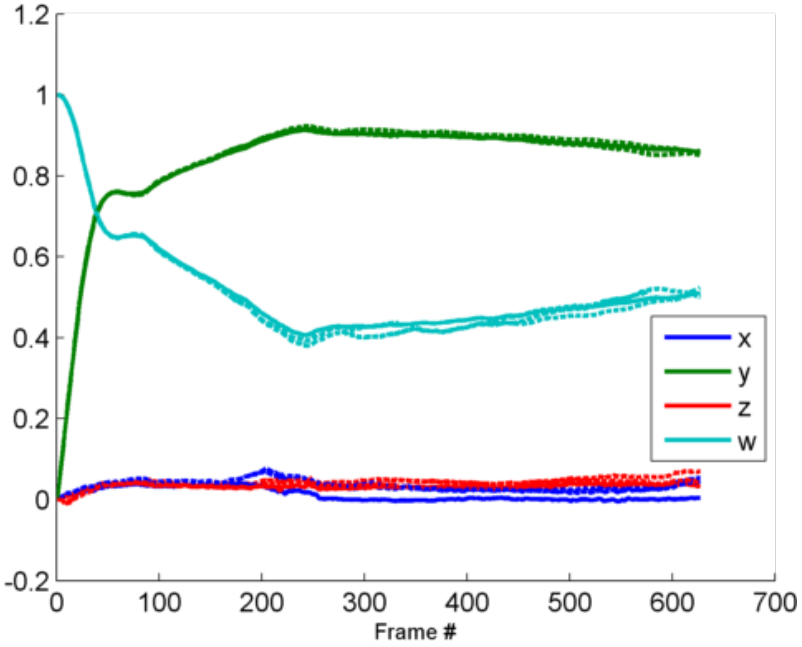


Figure 3. Pose results for kitti odometry clip 01 (Seq#2 in Table 2). The solid-lines are for the ground-truth pose values. The dotted-lines are for the estimated pose values from EKF single precision, EKF 64bit fixed-point, SRIF single precision and SRIF 32bit fixed-point implementations (in most cases overlapping with the ground-truth plots).

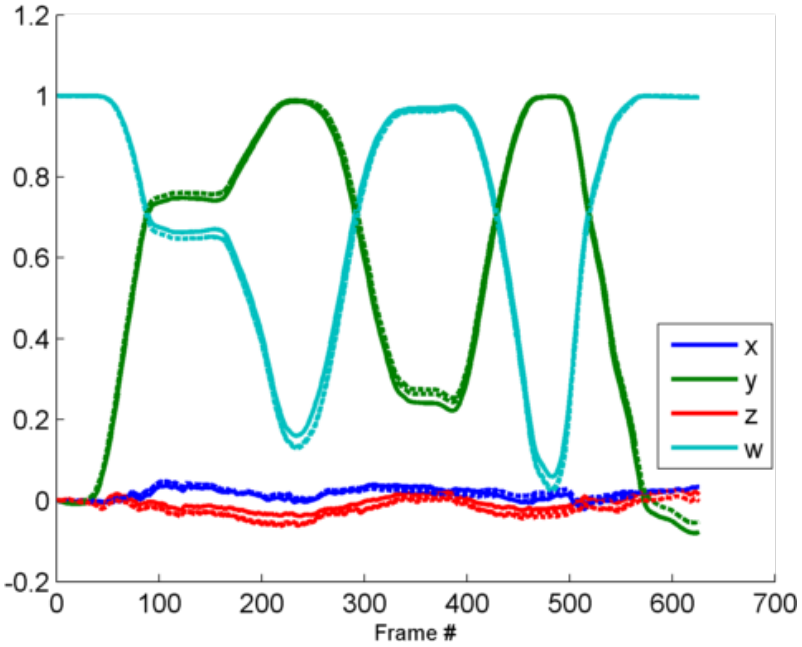


Figure 4. Pose results for kitti odometry clip 02 (Seq#3 in Table 2). The solid-lines are for the ground-truth pose values. The dotted-lines are for the estimated pose values from EKF single precision, EKF 64bit fixed-point, SRIF single precision and SRIF 32bit fixed-point implementations (in most cases overlapping with the ground-truth plots).

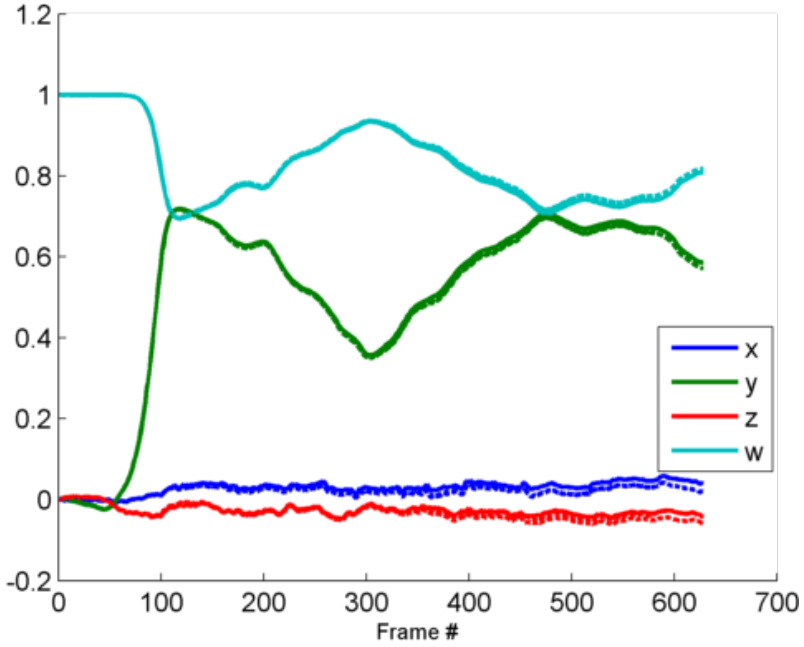


Figure 5. Pose results for kitti odometry clip 03 (Seq#4 in Table 2). The solid-lines are for the ground-truth pose values. The dotted-lines are for the estimated pose values from EKF single precision, EKF 64bit fixed-point, SRIF single precision and SRIF 32bit fixed-point implementations (in most cases overlapping with the ground-truth plots).

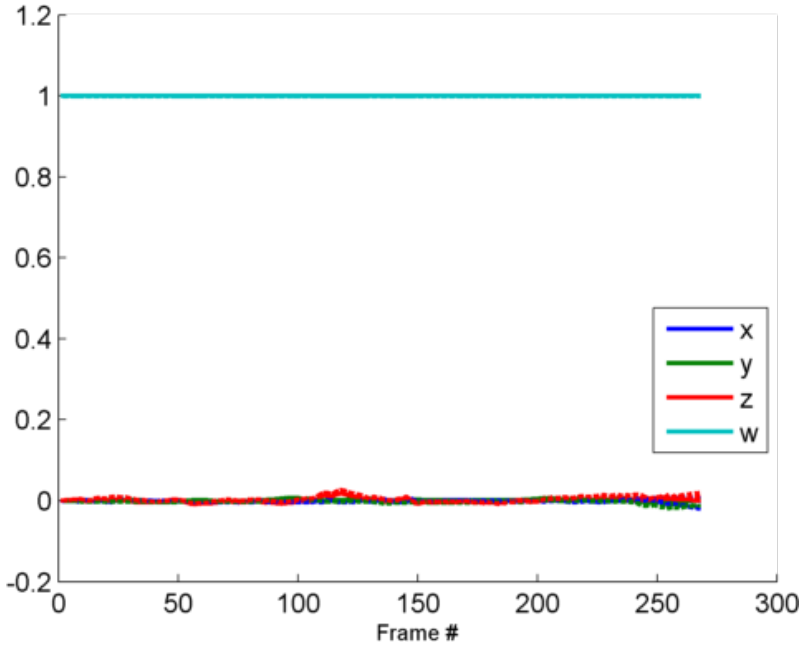


Figure 6. Pose results for kitti odometry clip 04 (Seq#5 in Table 2). The solid-lines are for the ground-truth pose values. The dotted-lines are for the estimated pose values from EKF single precision, EKF 64bit fixed-point, SRIF single precision and SRIF 32bit fixed-point implementations (in most cases overlapping with the ground-truth plots).

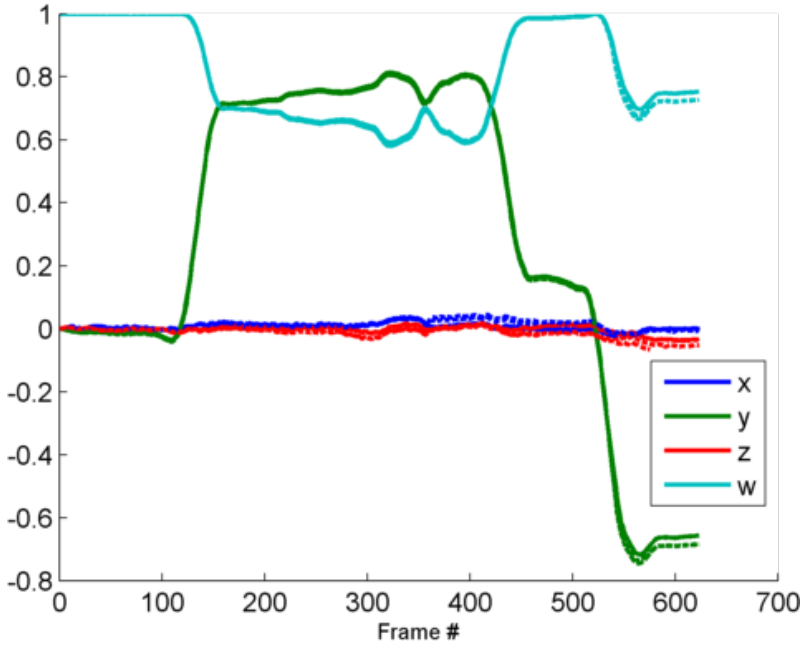


Figure 7. Pose results for kitti odometry clip 05 (Seq#6 in Table 2). The solid-lines are for the ground-truth pose values. The dotted-lines are for the estimated pose values from EKF single precision, EKF 64bit fixed-point, SRIF single precision and SRIF 32bit fixed-point implementations (in most cases overlapping with the ground-truth plots).

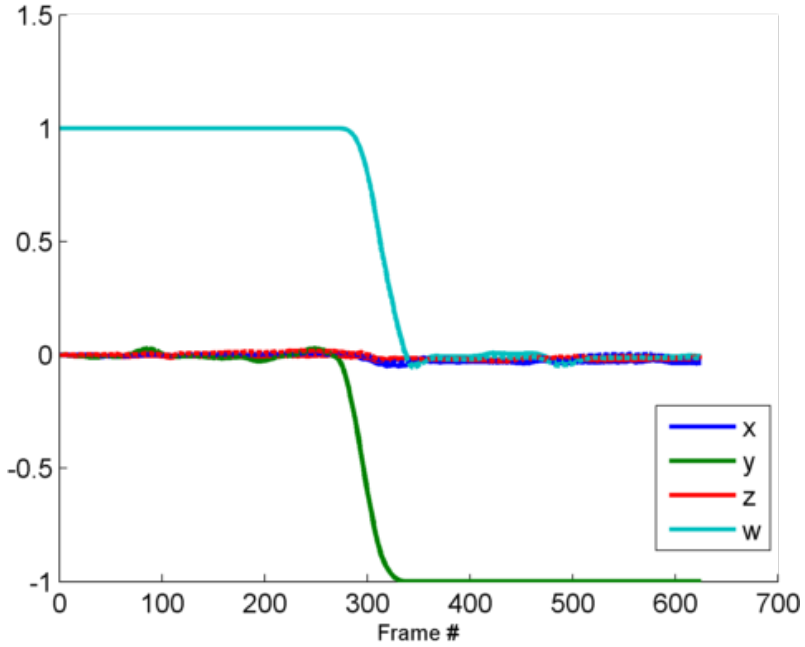


Figure 8. Pose results for kitti odometry clip 06 (Seq#7 in Table 2). The solid-lines are for the ground-truth pose values. The dotted-lines are for the estimated pose values from EKF single precision, EKF 64bit fixed-point, SRIF single precision and SRIF 32bit fixed-point implementations (in most cases overlapping with the ground-truth plots).

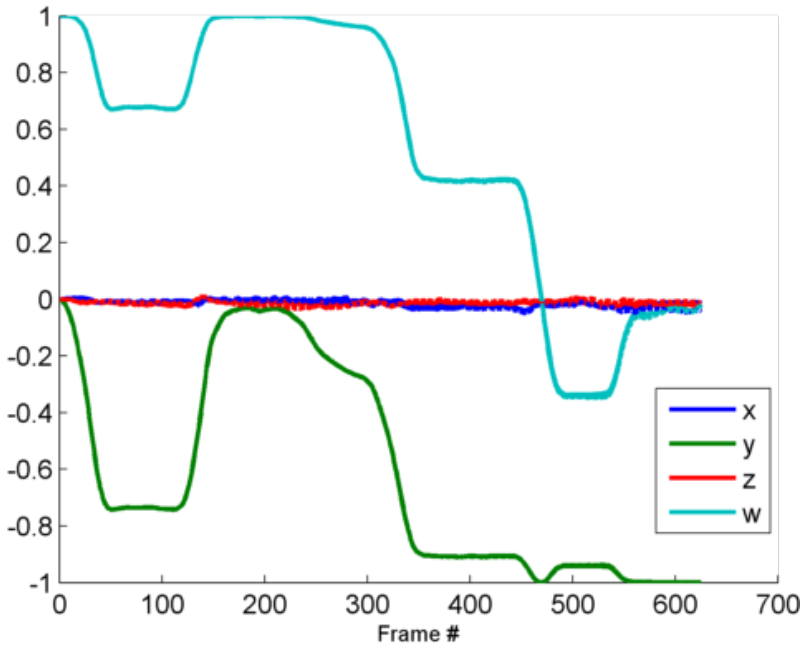


Figure 9. Pose results for kitti odometry clip 07 (Seq#8 in Table 2). The solid-lines are for the ground-truth pose values. The dotted-lines are for the estimated pose values from EKF single precision, EKF 64bit fixed-point, SRIF single precision and SRIF 32bit fixed-point implementations (in most cases overlapping with the ground-truth plots).

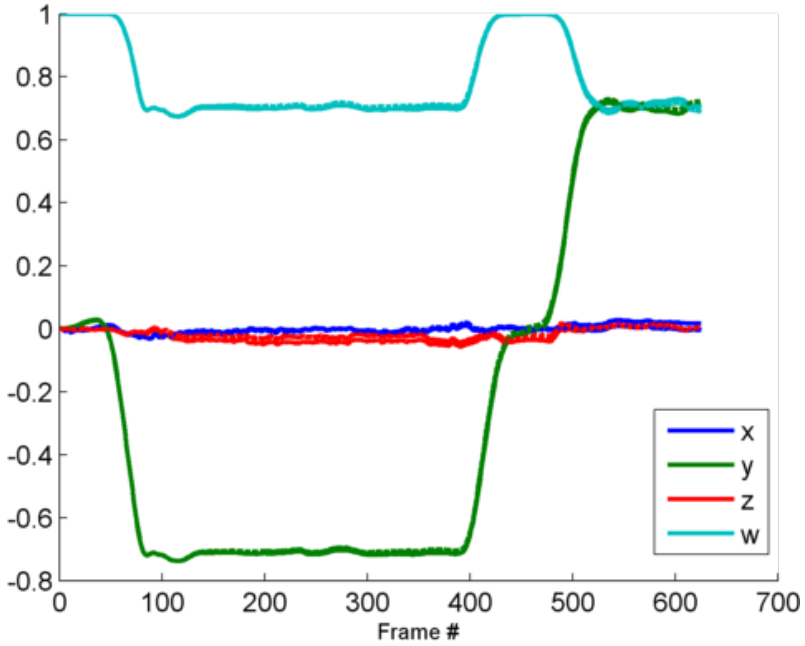


Figure 10. Pose results for kitti odometry clip 08 (Seq#9 in Table 2). The solid-lines are for the ground-truth pose values. The dotted-lines are for the estimated pose values from EKF single precision, EKF 64bit fixed-point, SRIF single precision and SRIF 32bit fixed-point implementations (in most cases overlapping with the ground-truth plots).

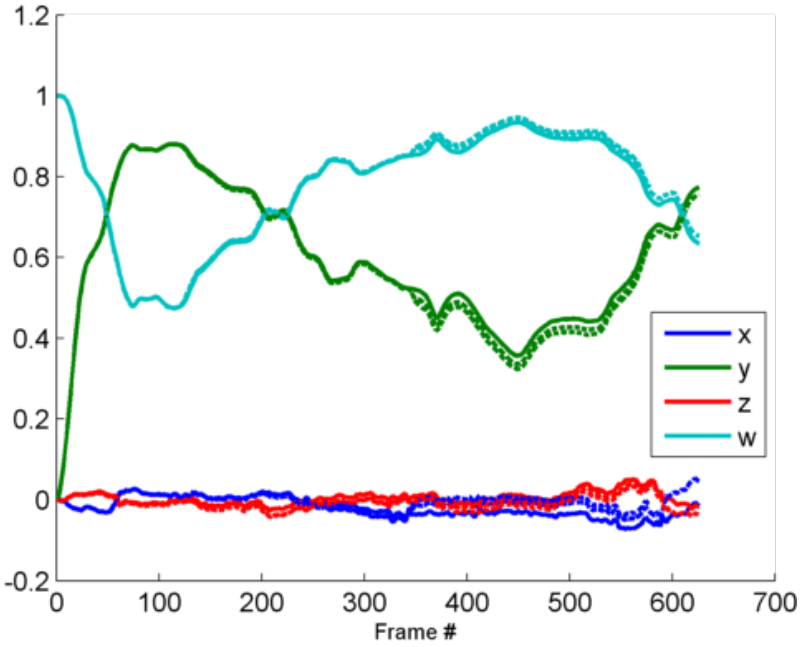


Figure 11. Pose results for kitti odometry clip 10 (Seq#10 in Table 2). The solid-lines are for the ground-truth pose values. The dotted-lines are for the estimated pose values from EKF single precision, EKF 64bit fixed-point, SRIF single precision and SRIF 32bit fixed-point implementations (in most cases overlapping with the ground-truth plots).