Inverse Depth EKF plus 1-Point RANSAC estimation

Javier Civera,
Oscar García
José M. Martínez Montiel

date: October 2009

1 Benchmark solution: Monocular EKF-SLAM

This section briefly describes the algorithm used to provide benchmark solutions to the datasets using monocular and wheel odometry streams. We begin by giving full detail of the algorithm used in next subsection; and present in the following one results obtained over the outdoor dataset Bovisa_2008-10-04. It should be noticed that the algorithm presented is based on novel contributions developed in the framework of the RAWSEEDS project and that constitute the state of the art in filtering monocular sequences. Specifically, the algorithm is based on the following publications: [Civera et al., 2008, Civera et al., 2009, Civera et al., 2010].

1.1 Camera-Centered EKF + 1-Point RANSAC

An illustrative scheme of the algorithm used is shown in algorithm 1, and is fully detailed along the subsection. The recent key filtering contributions that are combined in this scheme are: first, a camera-centered representation of the geometric entities in the estimation [Castellanos et al., 2007] that reduces the linearization error for long exploration trajectories, which will be the case in the large environments of the dataset. Second; inverse depth parametrization for point features [Civera et al., 2008] that will allow undelayed point initialization and low-parallax points mapping improving the accuracy of the estimation. Finally, a 1-Point RANSAC algorithm [Civera et al., 2009, Civera et al., 2010] will perform efficient outlier rejection based on filtering priors.

In the camera-centered representation, the estimation at every step k is parameterized as a multidimensional Gaussian $\mathbf{x}_k \sim \mathcal{N}\left(\hat{\mathbf{x}}_k, \mathbf{P}_k\right)$ that includes the location of the world reference frame $\mathbf{x}_W^C$ as a non-observable feature and the map $\mathbf{y}^C$, both in the current camera reference frame.

$$\hat{\mathbf{x}}_k^{C_k} = \begin{pmatrix} \hat{\mathbf{x}}_W^{C_k} \\ \hat{\mathbf{y}}^{C_k} \end{pmatrix}; \quad \mathbf{P}_k^{C_k} = \begin{pmatrix} \mathbf{P}_W^{C_k} & \mathbf{P}_{Wy}^{C_k} \\ \mathbf{P}_{yW}^{C_k} & \mathbf{P}_y^{C_k} \end{pmatrix} . \tag{1}$$

The map $\mathbf{y}^{C_k}$ is composed of n point features $\mathbf{y}_i^{C_k}$ which are parametrized using inverse depth coordinates as detailed in [Civera et al., 2008]:

$$\hat{\mathbf{y}}^{C_k} = \begin{pmatrix} \hat{\mathbf{y}}_1^{C_k} \\ \vdots \\ \hat{\mathbf{y}}_n^{C_k} \end{pmatrix}; \quad \mathbf{P}_y^{C_k} = \begin{pmatrix} \mathbf{P}_{y_1}^{C_k} & \cdots & \mathbf{P}_{y_1 y_n}^{C_k} \\ \vdots & \ddots & \vdots \\ \mathbf{P}_{y_n y_1}^{C_k} & \cdots & \mathbf{P}_{y_n}^{C_k} \end{pmatrix} . \tag{2}$$

Algorithm 1 Camera-Centered EKF + 1-Point RANSAC

```
INPUT: $\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}$ {EKF estimate at step $k-1$}
        $th$ {Threshold for low-innovation points.}
            {In this paper, $th = 1.0\ pixels$}
OUTPUT: $\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}$ {EKF estimate at step $k$}

{A. EKF prediction and individually compatible matches}
$[\hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}] = EKF\_prediction(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}, \mathbf{u})$
$[\hat{\mathbf{h}}_{k|k-1}, \mathbf{S}_{k|k-1}] = measurement\_prediction(\hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1})$
$\mathbf{z}^{IC} = search\_IC\_matches(\hat{\mathbf{h}}_{k|k-1}, \mathbf{S}_{k|k-1})$

{B. Get a reliable set of low-innovation inliers}
$\mathbf{z}^{li-inliers} = [\ ]$
for $i = 0$ to $n_{hyp}$ do
  $\mathbf{z}_i = select\_match(\mathbf{z}^{IC})$
  $\hat{\mathbf{x}}_i = EKF\_state\_update(\mathbf{z}_i, \hat{\mathbf{x}}_{k|k-1})$ {Notice: only state update; NO covariance update}
  $\mathbf{h}_i = predict\_all\_measurements(\hat{\mathbf{x}}_i)$
  $\mathbf{z}_i^{th} = find\_matches\_below\_a\_threshold(\mathbf{z}^{IC}, \mathbf{h}_i, th)$
  if $size(\mathbf{z}_i^{th}) > size(\mathbf{z}^{li-inliers})$ then
    $\mathbf{z}^{li-inliers} = \mathbf{z}_i^{th}$
  end if
end for

{C. Partial EKF update using low-innovation inliers}
$[\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}] = EKF\_update(\mathbf{z}^{li-inliers}, \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1})$

{D. Partial EKF update with high-innovation inliers}
$\mathbf{z}^{hi-inliers} = [\ ]$
for every match $j$ above a threshold $th$ do
  $[\mathbf{h}^j, \mathbf{S}^j] = point\_j\_prediction\_and\_covariance(\hat{\mathbf{x}}, \mathbf{P}, j)$
  $\nu^{\mathbf{j}} = \mathbf{z}^j - \mathbf{h}^j$
  if $\nu^{\mathbf{j}\top} \mathbf{S}^{\mathbf{j}^{-1}} \nu^{\mathbf{j}} < \chi^2_{\mathbf{2,0.01}}$ then
    $\mathbf{z}^{hi-inliers} = add\_match\_j\_to\_inliers(\mathbf{z}^{hi-inliers}, \mathbf{z}^j)$ {If individually compatible, add to inliers}
  end if
end for
if $size(\mathbf{z}^{hi-inliers}) > 0$ then
  $[\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}] = EKF\_update(\mathbf{z}^{hi-inliers}, \hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k})$
end if

{E. Composition step}
$[\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}] = composition(\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k})$
```

Location for the world reference frame with respect to the current camera frame is represented by its position vector and quaternion orientation

$$\hat{\mathbf{x}}_W^{C_k} = \begin{pmatrix} \hat{\mathbf{r}}_W^{C_k} \\ \hat{\mathbf{q}}_W^{C_k} \end{pmatrix} . \tag{3}$$

The algorithm integrating Extended Kalman Filter and 1-Point RANSAC outlier rejection can be divided in five stages, which are described below.

1.1.1 EKF Prediction and Individually Compatible Matches

For the prediction step at time k, the world reference frame and feature map are kept in the reference frame at time $k-1$ and a new feature that represents the motion of the sensor between $k-1$ and k is added:

$$\hat{\mathbf{x}}_{k|k-1}^{C_{k-1}} = \begin{pmatrix} \hat{\mathbf{x}}_W^{C_{k-1}} \\ \hat{\mathbf{y}}^{C_{k-1}} \\ \hat{\mathbf{x}}_{C_k}^{C_{k-1}} \end{pmatrix} \tag{4}$$

Predicted camera motion will be taken from the odometry measurements in the dataset; and predicted covariance will be obtained by linearizing the model and adding zero-mean Gaussian noise.

For each mapped feature $\hat{\mathbf{y}}_i^{C_k}$, its correspondence $\mathbf{z}_i$ will be searched in the 99% probability region from the predicted Gaussian pdf $\mathcal{N}\left(\hat{\mathbf{h}}_i, \mathbf{S}_i\right)$

$$\hat{\mathbf{h}}_i = \mathbf{h}_i\left(\hat{\mathbf{x}}_{k|k-1}^{C_{k-1}}\right) \tag{5}$$

$$\mathbf{S}_i = \mathbf{H}_i \mathbf{P}_{k|k-1}^{C_{k-1}} \mathbf{H}_i^\top + \mathbf{R}_i \ , \tag{6}$$

where $\mathbf{h}_i$ is the projection model –a pinhole camera with radial distortion as in [Civera et al., 2008] will be used–; $\mathbf{H}_i$ is the derivative of the projection model by the state vector and $\mathbf{R}_i$ is the covariance of the measurement noise.

This correspondence set, searched in the 99% probability region for each feature, is said to be individually compatible: that means, each match is compatible with the a priori model separately. But that does not imply that the whole set is jointly compatible; that is, compatible with the prior information when the whole set of matches is considered. There can exist outliers that are individually compatible –when evaluated separately from other matches– but not jointly compatible if they are evaluated with other matches. The algorithm described from here will extract from the initial set of individually compatible matches a jointly compatible set rejecting spurious data (see Fig. 1).

1.1.2 Selection of Low-Innovation Inliers Using 1-Point RANSAC

The hypothesize-and-verify loop follows here, and it is where the key difference with standard RANSAC arises. While standard RANSAC needs a certain number of points that depends on the particular problem to hypothesize a model, having prior knowledge coming from filtering will permit to propose hypothesis using only 1 data point. The advantage of requiring 1 point resides on the fact that it is easier to randomly select an spurious-free random sample; reducing then the number of samples and the associated computational cost.

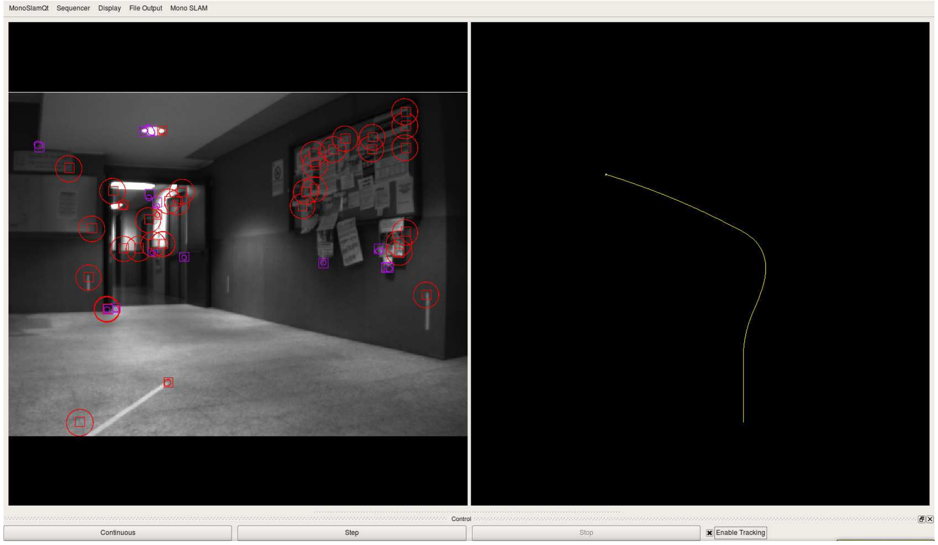

Figure 1: Capture of the Monocular SLAM system performing tracking of features on the sequence (left). A few set of predicted features (magenta) are rejected by 1-Point RANSAC while most of the features are correctly associated (red). Part of the estimated trajectory (right).

Another key aspect for the efficiency of the algorithm is that only a state vector update with one match is needed to compute an hypothesis, which is computationally cheap compared with the quadratic complexity of the whole EKF covariance update. This means that the hypothesis generation process will have in general a negligible cost compared with the standard EKF operations.

Following with the algorithm; support given by the matches for each hypothesis is computed simply counting the number of data points inside a heuristic threshold. Threshold should be chosen close to the measurement noise, which is what was done in this paper setting the value to 1 pixel.

1.1.3 Partial Update with Low-Innovation Inliers

Most supported hypothesis among the randomly constructed in previous step is selected. As threshold was chosen to be close to the measurement noise, all the matches $\mathbf{z}_i$ inside can be considered to be inliers having low innovation $\mathbf{z}^{li-inliers}$. The rest of the high-innovation measurements will be either high-innovation inliers, corresponding to either close points affected by translation or recently initialized points, or outliers.

A partial update of the covariance using this low innovation inliers will reduce most of the correlated priors in the Gaussian prediction. It is important to notice here that, while JCBB uses correlations between parameters to reject outliers, the approach of this paper first remove most of these correlations. After this removal, checking individual compatibility is enough to discard outliers.

1.1.4 Partial Update with High-Innovation Inliers

Individually compatible matches after the first partial update will be finally classified as inliers. Computation done in the first update serves as the starting point for the second partial update using rescued high-innovation inlier measurements. As in the first partial update, standard EKF update formulation will be applied.

It is important to remark here that, if it is true that dividing the covariance update step into two parts will introduce a extra computational overhead with respect to the case of a global update, such overhead will be of little importance compared with other computations of the filter –particularly covariance update, which is the most expensive computation.

1.1.5 Composition

After the update, a final composition step will be necessary in order to transform all the geometric entities in the filter from previous camera reference frame to the current one.The rigid transformation between the previous frame of reference and the current one is removed from the estimation. The resulting state vector is:

$$\hat{\mathbf{x}}_k^{C_k} = \begin{pmatrix} \hat{\mathbf{x}}_W^{C_k} \\ \hat{\mathbf{x}}_v^{C_k} \\ \hat{\mathbf{y}}^{C_k} \end{pmatrix} \quad , \tag{7}$$

where $\hat{\mathbf{x}}_W^{C_k}$, $\hat{\mathbf{x}}_v^{C_k}$ and $\hat{\mathbf{y}}^{C_k}$ have been computed by composition with the motion between frames $\hat{\mathbf{x}}_{C_k}^{C_{k-1}}$:

$$\begin{aligned} \hat{\mathbf{x}}_W^{C_k} &= \ominus \hat{\mathbf{x}}_{C_k}^{C_{k-1}} \oplus \hat{\mathbf{x}}_W^{C_{k-1}} && (8) \\ \hat{\mathbf{x}}_v^{C_k} &= \ominus \hat{\mathbf{x}}_{C_k}^{C_{k-1}} \oplus \hat{\mathbf{x}}_v^{C_{k-1}} && (9) \\ \hat{\mathbf{y}}^{C_k} &= \ominus \hat{\mathbf{x}}_{C_k}^{C_{k-1}} \oplus \hat{\mathbf{y}}^{C_{k-1}} \ . && (10) \end{aligned}$$

The final covariance is computed using the Jacobian of the composition equation $\mathbf{J}_{C_{k-1} \to C_k}$:

$$\mathbf{P}_k^{C_k} = \mathbf{J}_{C_{k-1} \to C_k} \mathbf{P}_k^{C_{k-1}} \mathbf{J}_{C_{k-1} \to C_k}^\top \ . \tag{11}$$

1.2 Results using Bovisa_2008-10-04 dataset

1.2.1 Methodology

In order to compare with RTK GPS, the trajectory provided by the filtering has to be previously aligned applying a rotation transformation over it

$$\begin{bmatrix} \mathbf{r}_{C_k}^W \\ 1 \end{bmatrix} = \begin{bmatrix} x_{C_k}^W \\ y_{C_k}^W \\ z_{C_k}^W \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{C_0}^W & \mathbf{t}_{C_0}^W \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} x_{C_k}^{C_0} \\ y_{C_k}^{C_0} \\ z_{C_k}^{C_0} \\ 1 \end{bmatrix} \ . \tag{12}$$

Translation offset $\mathbf{t}_{C_0}^W$ will be taken from the first GPS measurement, while rotation $\mathbf{R}_{C_0}^W$ will be obtained by minimizing the distance between the two trajectories –whole GPS and filtered trajectories– allowing a rotation motion.

Finally, the error of each camera position in the reconstructed path is computed as the Euclidean distance between each point of the estimated camera path and GPS path, both in the W reference,

$$e_k = \sqrt{\left(\mathbf{r}_{C_k}^W - \mathbf{r}_{GPS_k}^W\right)^\top \left(\mathbf{r}_{C_k}^W - \mathbf{r}_{GPS_k}^W\right)}. \tag{13}$$

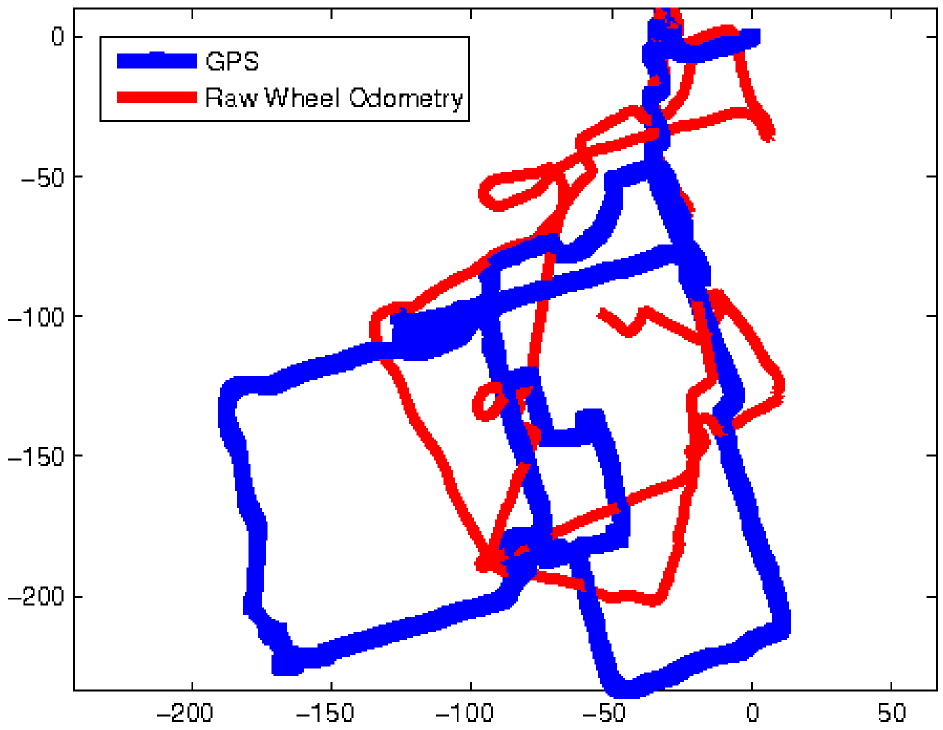

Figure 2: Raw odometry measurements (thin red) and GPS ground truth (thick blue). Errors of raw odometry are caused by early drift typical from proprioceptive sensors.

1.2.2 Monocular SLAM Results

The dataset chosen to provide a benchmark solution is Bovisa_2008-10-04. The length of the estimated trajectory is about 1310 meters long and was covered by RAWSEEDS mobile robot in 30 minutes being then the length of the sequence 54000 frames. Trajectory obtained from the combination of raw odometry and monocular information explained above is shown in figure 4; together with GPS ground truth for comparison. Maximum and mean error compared against GPS ground truth are 23.6 and 9.8 meters respectively.

Next figures are devoted to highlight the main inconveniences of raw odometry and monocular camera alone; and how the combination of the two sensors is able to overcome the drawbacks that both of them show separately. Figure 2 shows raw odometry lectures as a red thin line and GPS ground truth with a blue thick line for comparison. It can be observed that early drift appears and plotted trajectory is rather far from the ground truth value.

Figure 3 shows pure monocular estimation in thin red and GPS measurements in thick green. Observing carefully this plot, it can be observed that monocular camera is able to very accurately estimate orientation; but the unobservability of the scale produces drift in this parameter for the number of tracked features considered (25). Tracking a larger number of features as in [Civera et al., 2009] will reduce this scale drift, but real-time capabilities of the algorithm would be lost. Also, global scale is not recovered: in figure 3, scale had to be adjusted to a factor of 2.7 via a minimization process.

Finally, figure 4 details the estimated trajectory that can be achieved from the combination of the two sensors. It can be seen that problems commented in two previous figures disappear. An accurate estimation is achieved for a trajectory of 1.3 kilometers.

Figure 5 shows the histogram of the errors for the sequence. The number of tracked features

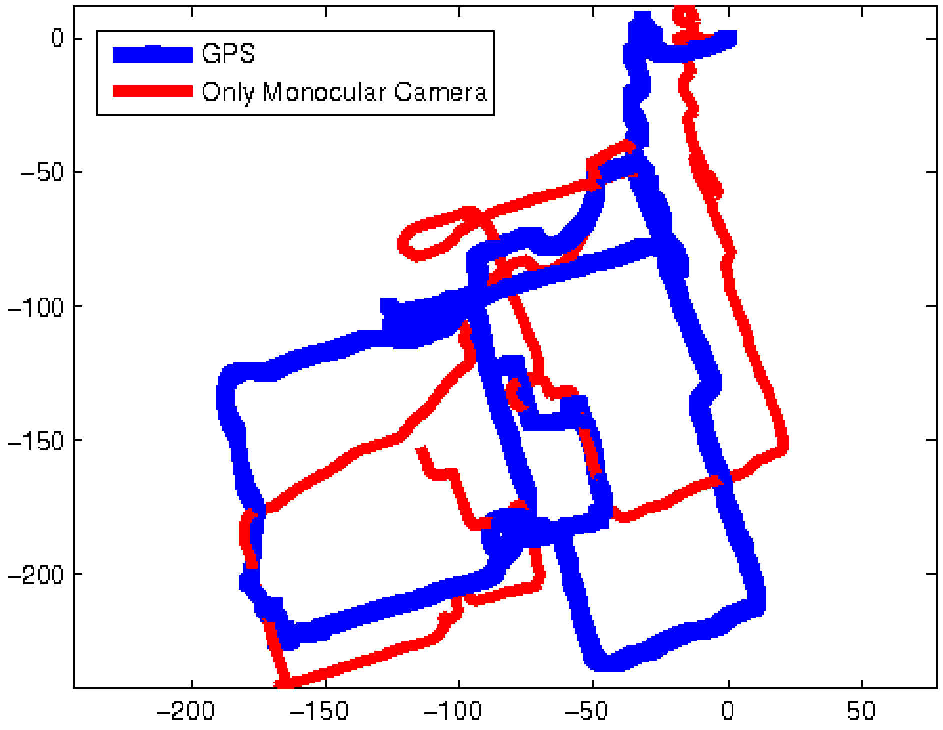

Figure 3: Pure monocular estimation (thin red) tracking 25 features and GPS ground truth (thick blue). Errors in this case are caused by scale drift, which is unobservable by a monocular camera.

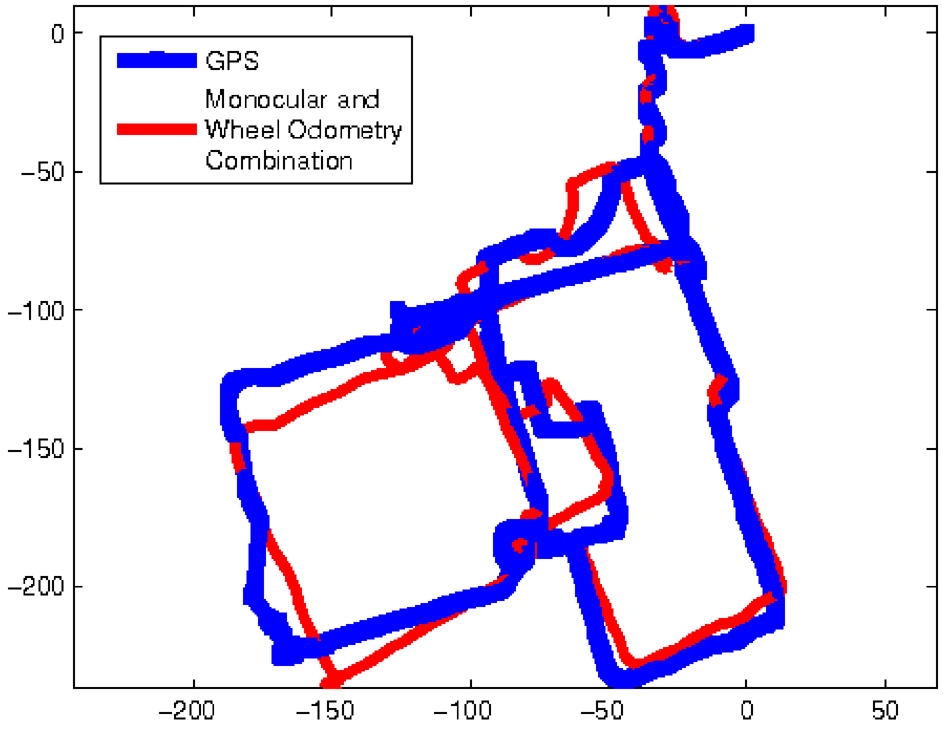

Figure 4: Monocular SLAM estimation from the combination of monocular camera plus wheel odometry (thin red) and GPS trajectory (thick blue).

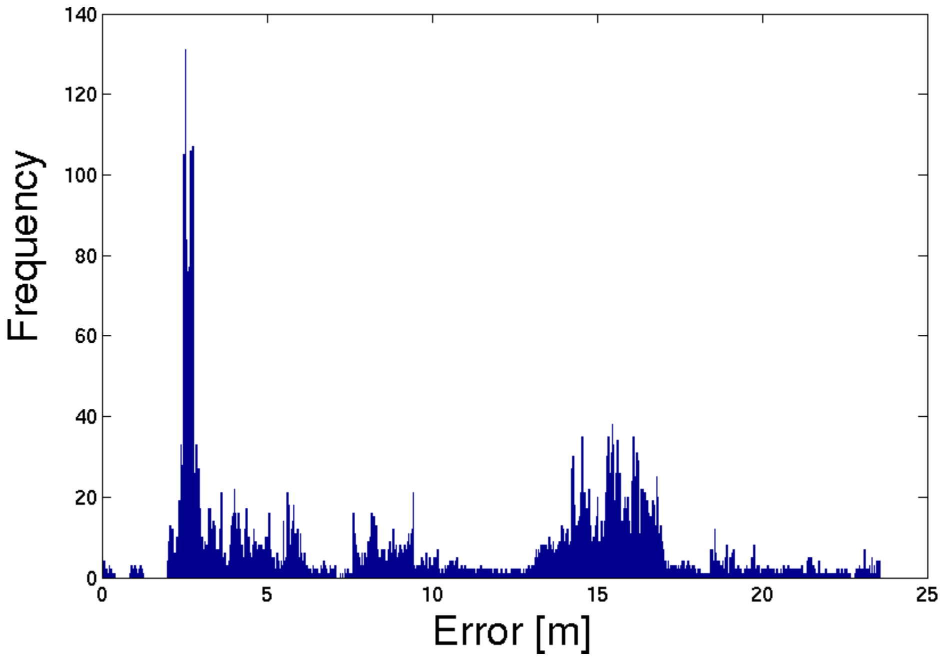

Figure 5: Histogram of the Monocular SLAM errors compared with GPS ground truth.

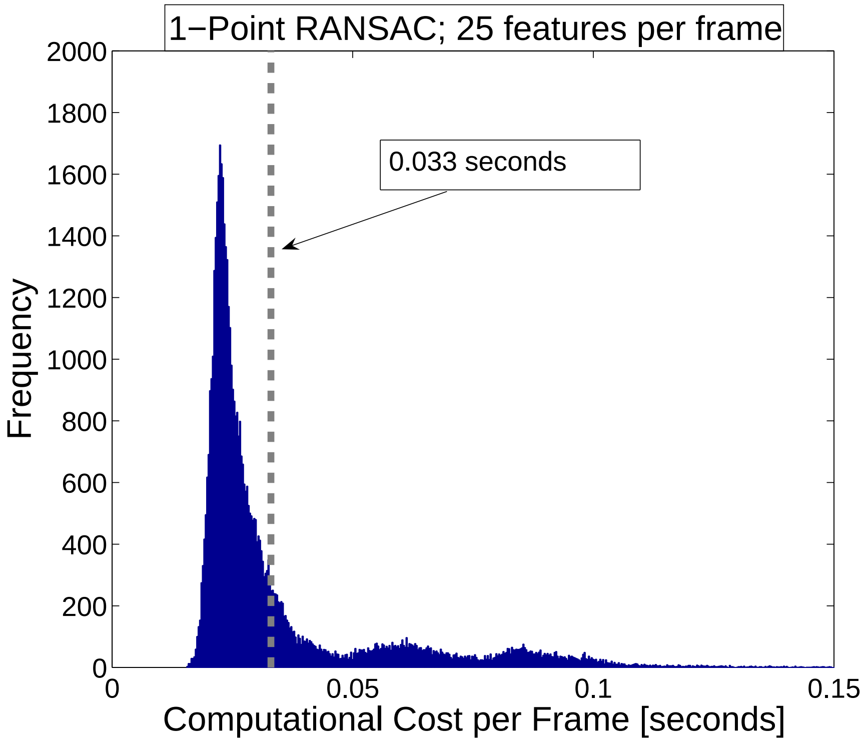

Figure 6: Histogram of the computational cost for 1-Point RANSAC when 25 image features are tracked.

per frame was kept in 25 for this experiment.

The processing time per frame for this experiment can be observed in figure 6 in the form of an histogram. It can be noticed that, although the proposed algorithm still does not entirely run at real time at 30 frames per second, it is really close: 70% of the frames are already under 33 miliseconds.

References

[Castellanos et al., 2007] Castellanos, J., Martinez-Cantin, R., Tardos, J., and Neira, J. (2007). Robocentric map joining: Improving the consistency of EKF-SLAM. *Robotics and Autonomous Systems*, 55(1):21–29.

[Civera et al., 2008] Civera, J., Davison, A. J., and Montiel, J. M. M. (2008). Inverse depth parametrization for monocular SLAM. *IEEE Transactions on Robotics*, 24(5):932–945.

[Civera et al., 2009] Civera, J., Grasa, O. G., Davison, A. J., and Montiel, J. M. M. (2009). 1-point RANSAC for EKF-based structure from motion. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

[Civera et al., 2010] Civera, J., Grasa, O. G., and Montiel, J. M. M. (2010). 1-point RANSAC for filtering. application to EKF-based visual odometry. Submitted to 2010 IEEE International Conference on Robotics and Automation; available on request.