Fast and Robust ℓ_1-averaging-based Pose Estimation for Driving Scenarios*

German Ros[1,2]
gros@cvc.uab.es
Julio Guerrero[3]
juguerre@um.es
Angel D. Sappa[1]
asappa@cvc.uab.es
Daniel Ponsa[1,2]
daniel@cvc.uab.es
Antonio M. López-Peña[1,2]
antonio@cvc.uab.es

[1] Computer Vision Center
Edifici O, Campus UAB, 08193
Bellaterra (Barcelona), Spain

[2] Computer Science Dept.
Universitat Autònoma de Barcelona
Campus UAB, Bellaterra (Barcelona), Spain

[3] Dept. de Matemática Aplicada
FIUM, Universidad de Murcia
Campus de Espinardo, Murcia, Spain

Abstract

Robust visual pose estimation is at the core of many computer vision applications, being fundamental for Visual SLAM and Visual Odometry problems. During the last decades, many approaches have been proposed to solve these problems, being RANSAC one of the most accepted and used. However, with the arrival of new challenges, such as large driving scenarios for autonomous vehicles, along with the improvements in the data gathering frameworks, new issues must be considered. One of these issues is the capability of a technique to deal with very large amounts of data while meeting the real-time constraint. With this purpose in mind, we present a novel technique for the problem of robust camera-pose estimation that is more suitable for dealing with large amount of data, which additionally, helps improving the results. The method is based on a combination of a very fast coarse-evaluation function and a robust ℓ_1-averaging procedure. Such scheme leads to high-quality results while taking considerably less time than RANSAC. Experimental results on the challenging KITTI Vision Benchmark Suite are provided, showing the validity of the proposed approach.

1 Introduction

Robust camera-pose estimation is a fundamental stage of many computer vision problems, being specially important for Visual Simultaneous Localization and Mapping (*VSLAM*) and Visual Odometry (*VO*) systems. Both problems have received a relevant amount of attention during the last decades, e.g., [5][20][22][26]; professing a special dedication to algorithms capable of dealing with high levels of noise and outliers, such as [3][6].

When camera-pose estimation is applied as a part of a *VSLAM-VO* framework, it is mandatory to consider the real-time constraint and how this stage affects the overall performance of the system. State-of-the-art approaches commonly address this problem by making use of the well-known methodology proposed in RANSAC [6] or any of its variants

*This work has been supported by the Universitat Autònoma de Barcelona, the Fundación Séneca 08814PI08 and the Spanish government, by the projects FIS201129813C0201; TRA201129454C0301 (eCo-DRIVERS); TIN201125606 (SiMeVé) and TIN201129494C0302 (FireWATCHER)

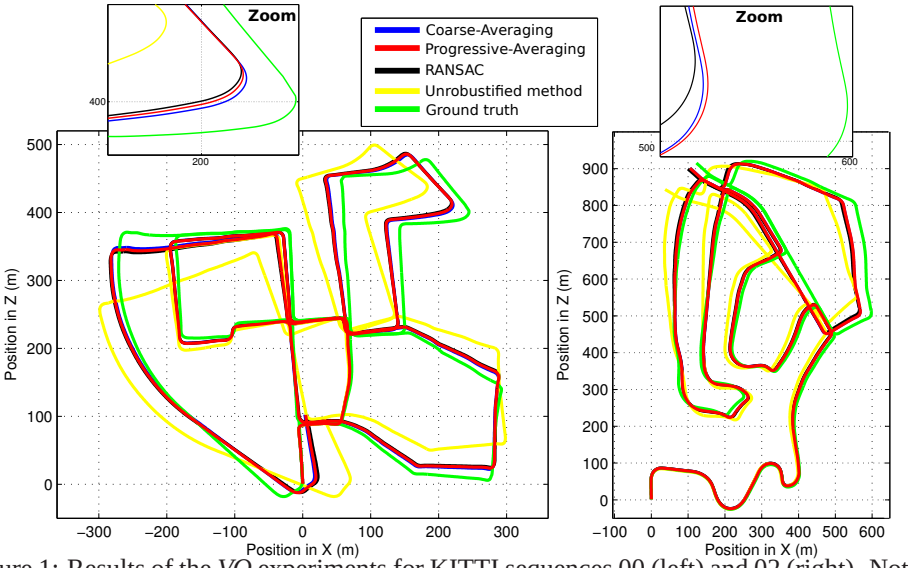

Figure 1: Results of the *VO* experiments for KITTI sequences 00 (left) and 02 (right). Notice that, although all robust methods lead to similar trajectories, C-Avg and P-Avg remain closer to the ground truth.

[1][23]. This usually leads to good results, but there are cases in which the performance of these algorithms drastically decreases. One of these cases takes place when the amount of input information, constraining the affected pair of views, is too large. This phenomenon is present in modern frameworks, such as [10][9], which are able to generate thousands of correspondences between pairs of monocular images or the four views of a moving stereo-rig, at two time instants, while performing in real-time in a standard CPU. RANSAC-like methods are affected by this "excess" of information, what produces an increment on the time dedicated to evaluate and rank the generated models. In order to avoid this drawback, real-time implementations opt to use just a part of the available data, therefore discarding a great amount of information and penalizing the accuracy of the resultant models.

In this paper, we propose a novel alternative to the problem of robust pose estimation with application to *VO* systems in large driving scenarios. Our approach is designed to deal with large amounts of data in a very efficient way and we show that such a property helps improving estimation results. The algorithm is based on a combination of coarse model evaluation along with a posterior stage of robust ℓ_1-averaging. We show how our technique leads to similar or better results than those produced by RANSAC, while performing in less time. Additionally, our experiments suggest that this sort of strategy is very suitable for large urban environments, where rich textures are easily found and inliers ratio is high. In all our tests we use the KITTI Vision Benchmark Suite, a novel benchmark presenting challenging urban scenarios [11]. Additionally, we provide an efficient implementation of our approach[1].

The remainder of this paper is structured as follows. Section 2 contextualizes our approach according with the literature. Then, in section 3, the method and its constitutive stages are described. Section 4 introduces a modified version of the model generation stage that leads to better results. We validate these concepts throughout real data tests in section 5. Finally, we summarize our findings in section 6, giving an advance of our future work.

[1] code available at: https://github.com/germanRos/l1avgvo

2 Related Work

Robust pose estimation techniques were originally inspired by the extensive work done within the field of statistics [14]. Techniques such as the Huber robust M-estimator [18] are still widely used for this purpose, giving rise to *VSLAM-VO* frameworks like the proposed by Comport *et al*. [4]. The principal advantage of M-estimators is its algorithmic simplicity, which in practice means a good trade-off between robustness and computational efficiency. However, a negative aspect of these tools is that, by design, they have to be applied to each individual association independently. This prevents applying data reduction techniques [15], which have proven to be very attractive for achieving real-time capabilities.

On the other hand, a large part of the literature about robust methods for motion estimation is centred on consensus techniques such as RANSAC [6]. After more than thirty years, RANSAC is still one of the most outstanding methods and resides at the core of many state-of-the-art *VSLAM-VO* frameworks [26][27], since it produces good results and is simple. Many variations of the original scheme have been proposed in order to mitigate known drawbacks [1][23]. It is out of the scope of this paper to review the advantages of such techniques, but we must highlight two remarkable variations that have inspired this work.

The first of these methods is Progressive Sample Consensus (PROSAC), proposed by Chum and Matas in [2]. The idea of PROSAC is to benefit from the information generated by visual matching procedures, since for each matched keypoints it is possible to assign a score that act as a vague prior of the association quality. This extra information is used to sort the set of matches and to impose an order in the model generation step. In this way, matches with good scores are more likely to be drawn earlier, which usually improves the creation of high quality models with less effort. In both approaches, the evaluation time for each model is still dominated by the size of the input data, which is an important drawback in the context of *VSLAM-VO* applications.

To solve this issue, Nistér proposed Preemptive RANSAC [21]; a technique that follows a breadth-first scheme to evaluate hypotheses with incremental data. This leads to a fast rejection of some models, what reduces the overall computation. In the presence of large and "clean" amounts of data, the possible configurations of Preemptive RANSAC can lead to two undesirable situations: (i) due to the large amount of good matches too many good models are kept across the hierarchy, producing an overload of the evaluation stage; (ii) a restrictive breadth-first search is used and a large part of the information is never considered.

A radically different strategy was proposed by Govindu [13][12], who explored the idea of combining multiple camera poses (hypotheses) in a coherent fashion by using averaging techniques on a manifold. The algorithm proceeds by generating several poses from different subsets of data and then through averaging it tries to create a robust final pose. These works laid the foundations of pose averaging and inspired several new techniques. However, the robustness of these averaging methods might be compromised when a fraction of the poses are severely affected by noise. In order to make pose averaging more robust, Hartley *et al*. [16][17] proposed the use of pose averaging under the ℓ_1-norm, also known as the geometric median. That change greatly improves the robustness of the averaging, leading to very accurate results. This idea is validated with $\mathbb{SO}(3)$ models for Essential matrix estimation.

The method here proposed is inspired by [2][12][16][21], but presents clear differences with them. First of all, we focus on ℓ_1-averaging on $\mathbb{SE}(3)$, being our main target stereo *VO* for large urban environments, a domain in which these techniques have not been previously tested. Furthermore, our approach can produce robust results from large amounts of data in less than 15 ms, which makes it much faster than RANSAC.

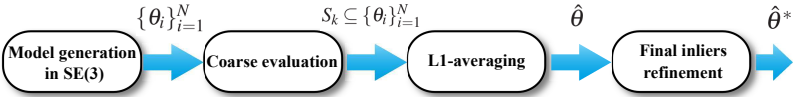

Figure 2: Main pipeline stages of the presented technique.

3 Robust Pose Estimation via ℓ_1-averaging

We have focused the development of our method on producing fast and robust camera-pose estimation for *VSLAM-VO* applications. It is assumed that the input sensor is a fully calibrated stereo-rig and that pixel correspondences between the four views are provided (for instance, by using [10]). We model motion as being rigid transformations in the 3D space, i.e., as elements of the group $\mathbb{SE}(3)$.

The algorithm here explained follows a scheme of model generation and evaluation, in the same line as RANSAC. However, the kind of evaluation proposed differs from the former. RANSAC makes use of a robust evaluation function $F_{inliers}$ that counts the number of inliers of a given model θ_i. The model with greatest number of inliers $\hat{\theta} = argmax_i\{F_{inliers}(\theta_i)\}_{i=1}^N$ is conserved as the best candidate. This strategy can be computationally expensive for real-time purposes when the amount of correspondences is very large, e.g., of the order of thousands, a number that is becoming common in modern acquisition frameworks (e.g., [8][10]).

Our proposal consists in changing the evaluation stage by introducing a new cost function that can be used to score models in constant time, independently of the data size. We call this kind of functions F_{coarse} since it performs a very quick evaluation but at the price of producing less reliable assessments. Actually, the output of this function cannot be directly used to select the best candidate, but it can be used to select a subset of the models $S_k \subseteq \{\theta_i\}_{i=1}^N$ containing a very high proportion of "good" models (those with a similar amount of inliers as $\hat{\theta}$). Then, the models in the subset S_k are combined in a robust way by using ℓ_1-averaging on $\mathbb{SE}(3)$. The idea is that both coarse evaluation and model averaging can be performed extremely quickly, giving rise to a fast and robust estimation technique. This strategy is summarized in the diagram of Fig. 2.

3.1 Model generation

This initial stage consists in generating N models $\theta_i \in \mathbb{SE}(3)$ from the available data $\mathcal{X} = \{(x_{l,p}, x_{r,p}, x_{l,c}, x_{r,c})^{(i)}\}_{i=1}^D$. Here, $x_{j,k} = (u_{j,k}, v_{j,k})$ stands for pixel coordinates, the subscript $j = \{l, r\}$ describes the pixel camera source (l)eft or (r)ight and the subscript $k = \{p, c\}$ specifies if the pixel comes from the (p)revious or the (c)urrent frame. Each model is generated by randomly drawing a minimal number of matches $M \leq D$ to constraint the model; in our case, since we work in $\mathbb{SE}(3)$, $M = 3$ matches. Given that the stereo-rig is fully calibrated the method starts by triangulating all the 3D points $\{X_{l,p}^{(i)}\}_{i=1}^D$ such that:

$$\Pi_3 \left(\begin{bmatrix} 1 & 0 & 0 & -c_u \\ 0 & 1 & 0 & -c_v \\ 0 & 0 & 0 & f \\ 0 & 0 & 1/B & 0 \end{bmatrix} \begin{bmatrix} u_{l,p} \\ v_{l,p} \\ d_p \\ 1 \end{bmatrix}^{(i)} \right) = X_{l,p}^{(i)} \tag{1}$$

Here $u_{l,p}$ and $v_{l,p}$ are the components of the pixel in the left previous view and $d_p = u_{l,p} - u_{r,p}$ is its disparity with respect to the left camera. It is assumed that both cameras share the same focal length f and the principal point (c_u, c_v), having no skew term. B is the stereo-rig baseline and $\Pi_i : \mathbb{P}^i \subseteq \mathbb{R}^{i+1} \rightarrow \mathbb{R}^i$ is a standard projection function. Afterwards, each model is generated by optimizing the cost function shown in Eq. 2, which represents an algebraical

cost, chosen to produce a behaviour similar to the reprojection error of 3D points in the current views. Here $\hat{X}^{(i)}$ and $\hat{x}^{(i)}$ are homogeneous 3D and 2D points, respectively, while K is the standard 3×3 matrix of intrinsic parameters and $\vec{B}$ stands for the vector $[B,0,0]^T$.

$$C(\psi)=\sum_{i=1}^{M}\left\|\mathrm{K}\,\Pi_3\left(\mathbf{exp}_r(\psi)\hat{X}_{l,p}^{(i)}\right)\times\hat{x}_{l,c}^{(i)}\right\|_{\ell_2}^2+\left\|\mathrm{K}\,\left(\Pi_3\left(\mathbf{exp}_r(\psi)\hat{X}_{l,p}^{(i)}\right)-\vec{B}\right)\times\hat{x}_{r,c}^{(i)}\right\|_{\ell_2}^2 \quad (2)$$

The optimization is carried out on the $\mathbb{SE}(3)$ manifold by considering a minimal parametrization ψ along with a first order retraction $\mathbf{exp}_r$. Such a retraction is a first order approximation of the actual exponential map, which maps from the Lie algebra to the manifold to ensure all the constraints of the group are met. Furthermore, the retraction is simpler to compute than the exponential map and there is no loss of accuracy; in this case the retraction used is the Cardan map (Eq. 3).

$$\mathbf{exp}_r(\psi)=\begin{bmatrix}\cos\psi_2\cos\psi_3 & -\cos\psi_2\cos\psi_3 & -\sin\psi_2 & \psi_4\\ \cos\psi_1\sin\psi_3-\sin\psi_1\sin\psi_2\sin\psi_3 & \cos\psi_1\cos\psi_3+\sin\psi_1\sin\psi_2\sin\psi_3 & -\sin\psi_1\cos\psi_2 & \psi_5\\ \sin\psi_1\sin\psi_3+\cos\psi_1\sin\psi_2\cos\psi_3 & \sin\psi_1\cos\psi_3-\cos\psi_1\sin\psi_2\sin\psi_3 & \cos\psi_1\cos\psi_2 & \psi_6\\ 0 & 0 & 0 & 1\end{bmatrix} \quad (3)$$

The cost function in Eq. 2 is optimized with some iterations of the Levenberg–Marquardt algorithm to produce the model $\theta=\begin{bmatrix}R & T\\ \vec{0} & 1\end{bmatrix}=\mathbf{exp}_r(\mathrm{argmin}_\psi C(\psi))$, where R is a rotation matrix and T a translation vector. In a typical configuration we generate between $N=100$ and $N=2000$ models.

3.2 Coarse Evaluation

Each of the models is evaluated with a so-called coarse function F_{coarse}. This function has been designed to be extremely fast, an objective that is achieved thanks to the use of a *Reduced Measurement Matrix* (*RMM*) [15][25]. *RMMs* are algebraical reductions of the input data $\mathcal{X}$ that create a compact equivalent $\boldsymbol{M}$ under the ℓ_2-norm. The advantage of this reduction is that $\boldsymbol{M}$ can be computed very efficiently even for very large collections of data and this has to be done just once, at the beginning of the process. Eq. 4 shows the structure of F_{coarse} and how to form $\boldsymbol{M}$ for the cost function defined in Eq. 2:

$$F_{coarse}(\theta)=\sum_{i=1}^{D}\left\|\mathrm{K}\,\Pi_3\left(\theta\hat{X}_{l,p}^{(i)}\right)\times\hat{x}_{l,c}^{(i)}\right\|_{\ell_2}^2+\left\|\mathrm{K}\,\left(\Pi_3\left(\theta\hat{X}_{l,p}^{(i)}\right)-\vec{B}\right)\times\hat{x}_{r,c}^{(i)}\right\|_{\ell_2}^2=$$
$$\sum_{i=1}^{D}\left\|W_l^{(i)}\breve{\theta}\right\|_{\ell_2}^2+\left\|W_r^{(i)}\breve{\theta}\right\|_{\ell_2}^2=\left\|\boldsymbol{W_l}\breve{\theta}\right\|_{\ell_2}^2+\left\|\boldsymbol{W_r}\breve{\theta}\right\|_{\ell_2}^2=\breve{\theta}^T\underbrace{\boldsymbol{W_l}^T\boldsymbol{W_l}}_{\boldsymbol{M_l}}\breve{\theta}+\breve{\theta}^T\underbrace{\boldsymbol{W_r}^T\boldsymbol{W_r}}_{\boldsymbol{M_r}}\breve{\theta}= \quad (4)$$
$$\breve{\theta}^T(\boldsymbol{M_l}+\boldsymbol{M_r})\breve{\theta}$$

here, $\breve{\theta}=[\mathbf{stack}(R),\mathbf{stack}(T),1]^T$, i.e., a stacked version of θ with an homogeneous component. The key terms of this expression are the 3×13 matrices $W_l^{(i)}$ and $W_r^{(i)}$, which have the following structure:

$$W_j^{(i)}=\begin{bmatrix}[0]_{1\times 3} & fX_{l,p}^{(i)^T} & (c_v-v_j^{(i)})X_{l,p}^{(i)^T} & 0 & f & (c_v-v_j^{(i)}) & 0\\ -fX_{l,p}^{(i)^T} & [0]_{1\times 3} & (u_j^{(i)}-c_u)X_{l,p}^{(i)^T} & -f & 0 & (u_j^{(i)}-c_u) & \alpha\\ fv_j^{(i)}X_{l,p}^{(i)^T} & -fu_j^{(i)}X_{l,p}^{(i)^T} & (c_uv_j^{(i)}-c_vu_j^{(i)})X_{l,p}^{(i)^T} & fv_j^{(i)} & -fu_j^{(i)} & (c_uv^{(i)}-c_vu_j^{(i)}) & \beta\end{bmatrix}_{j=l,r} \quad (5)$$

For $W_r^{(i)}$, $\alpha=Bf$ and $\beta=-Bfv_r^{(i)}$, while both are zero for $W_l^{(i)}$. As a simplification of the notation $u_j^{(i)}$ and $v_j^{(i)}$ are used for denoting the pixel components of the i-th correspondence

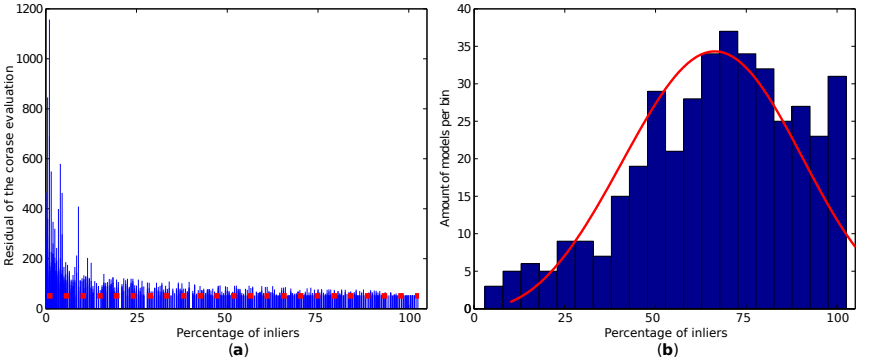

Figure 3: **(a)** relationship between the percentage of inliers of a model and its F_{coarse} residual; the red dotted line shows the residual for the best model (in terms of inliers). **(b)** histogram showing the distribution of models with a specific percentage of inliers after the selection.

in the current left or right view. Finally, each $\boldsymbol{W_j}$ is the stacking of all the $W_j^{(i)}$ blocks, forming up a $3D \times 13$ matrix. The resultant $\boldsymbol{M}$ is a 13×13 matrix that can be considered as a "condensed" ℓ_2-norm equivalent of Eq. 2. The main drawback of this compact version is that outliers cannot be easily detected any longer, and therefore, the residual $e_i = F_{coarse}(\theta_i) = \check{\theta}_i^T \boldsymbol{M} \check{\theta}_i$ should not be considered as an indicator of the goodness of a model.

After a thorough analysis of real data sequences, we observed that models with a high number of inliers (the expected good models) produce low residuals for F_{coarse}, even when $\boldsymbol{M}$ contains outliers. On the other hand, models corresponding with a low number of inliers present random values for F_{coarse}, producing low residuals just occasionally. Such a phenomenon is shown in Fig. 3 **(a)**. This property is strong enough to allow for an ordering of $\{\theta_i\}_{i=1}^N$. It turns out that, by selecting the N_k models with lowest residual —the selection criterion used for this approach— the number of "good" models tends to be much higher than the number of "bad" models (those with low proportions of inliers). This is shown in Fig. 3 **(b)**, for which we selected $N_k = 500$ models out of $N = 1000$ based on F_{coarse} and created a histogram with the frequency of models for each given amount of inliers. It is evident from the histogram that the distribution is skewed to the right (i.e., models with high ratio of inliers). As we discuss in next section, such a distribution can be exploited by a method of robust averaging as long as the proportion of good models stays above the 50%. Later, we will show how the correct ordering of matches helps to fulfill this condition.

3.3 Robust Averaging and Pose Refinement

In this stage a new high-quality model is generated from the information encoded in $S_k \subseteq \{\theta_i\}_{i=1}^N$. Instead of trying to pick up the best candidate from S_k we opted for combining them all with a pose-averaging method [13][17]. The main reason is that this operation can be done extremely quick in modern computers, taking barely one millisecond for an amount of 500 models. However, classical ℓ_2-averaging methods are not robust, leading to wrong results when S_k is partially corrupted. To avoid this drawback we make use of a robust ℓ_1-averaging method proposed by Hartley *et al*. [16]. ℓ_1-averaging uses the multi-dimensional equivalent of the median operator and can stand up to a 50% of corruption in S_k, something that in practice is achieved due to the mentioned phenomenon.

Each of the models in S_k represents a rigid transformation in 3D, i.e., $\theta_i \in \mathbb{SE}(3)$. Therefore, the final averaged model $\hat{\theta}$ must be an element of $\mathbb{SE}(3)$ as well. To enforce this

constraint the method makes use of the Lie group properties of $\mathbb{SE}(3)$ and its associated Lie algebra $\mathfrak{se}(3)$. In this way, the averaging is performed by projecting each model θ_i to the tangent space of the current estimate of $\hat{\theta}$ in $\mathfrak{se}(3)$, through the logarithm map se3Log. Such a space is isomorphic with $\mathbb{R}^6$ and in consequence 6-vectors ψ can be used as a local representation. Then, the Weiszfeld algorithm is applied to iteratively estimate the geodesic median of the 6-vectors, mapping the result back to $\mathbb{SE}(3)$ through the exponential map se3Exp. The authors in [16] propose to use retractions to approximate both exponential and logarithm maps, but in this stage we make use of the actual maps, as defined in [7]. For the sake of completeness a summary of the procedure is showed in Algorithm 1, although interested readers are referred to [16][17] for further details.

The initialization of this method can be done with a random guess or by using a non-robust ℓ_2-averaging method, as suggested by Hartley *et al*. [16]. Since S_k contains just a few wrong elements, the procedure keeps in the domain of the exp and log maps. As an optional step after the averaging, the final model $\hat{\theta}^*$ is refined by computing the inliers set of $\hat{\theta}$ as in RANSAC to perform a final optimization with the cost function defined in Eq. 2.

Algorithm 1 ℓ_1-averaging with the Weiszfeld algorithm

$\hat{\theta} \leftarrow$ Initial guess (ℓ_2-averaging [12])
repeat
  $\psi_i \leftarrow \text{se3Log}(\theta_i \hat{\theta}^{-1})$, for $i = 1, \ldots, N_k$
  $\delta \leftarrow \frac{\sum_{i=1}^{k} \psi_i / \|\psi_i\|_{\ell_2}}{\sum_{i=1}^{k} 1 / \|\psi_i\|_{\ell_2}}$
  $\hat{\theta} \leftarrow \text{se3Exp}(\delta)\ \hat{\theta}$
until $\|\delta\|_{\ell_2} < \varepsilon$
return $\hat{\theta}$

4 Progressive Sampling Scheme

In previous sections we stated that the use of F_{coarse} along with ℓ_1-averaging lead to robust model estimations in the presence of outliers. It is also claimed that this is true as long as the ratio of good models in S_k remains above the 50%. According to our experience, when the target problem is $VSLAM-VO$ in urban environments (outdoors), the data acquired by state-of-the-art frameworks tends to be quite good. This fact is a consequence of the rich textures present in urban scenarios and it favours the correct behaviour of estimation methods.

However, there are also situations where the quality of the matches is very poor, e.g., highways scenes. For these cases the creation of a suitable set S_k requires to generate a very high number of models —usually up to tens of thousands. This problem stems from the need of exploring a large part of the correspondences until suitable models are generated. If the process is stopped too early, the method would end up selecting k near-random models and the final one would be of no value. Therefore, in difficult scenes (scenes with a low ratio of inliers $\tau < 40\%$), the model generation stage becomes the bottle-neck of our approach.

We found in our experiments that this issue can be greatly reduced by substituting the random draw of matches for a priority scheme that favours the generation of better models. This is the principle proposed in PROSAC and consists in using information about the quality of the correspondences as a prior of their goodness. For visual matching this information is already computed and have proven to be a good prior (see section 5). The changes in the model generation scheme required to reflect this policy are straight forward. Firstly, the correspondences in $\mathcal{X}$ are sorted according to their scores, giving rise to $\mathcal{X}_s = \{\mathcal{X}_{s,1}, \ldots, \mathcal{X}_{s,D}\}$. Then,

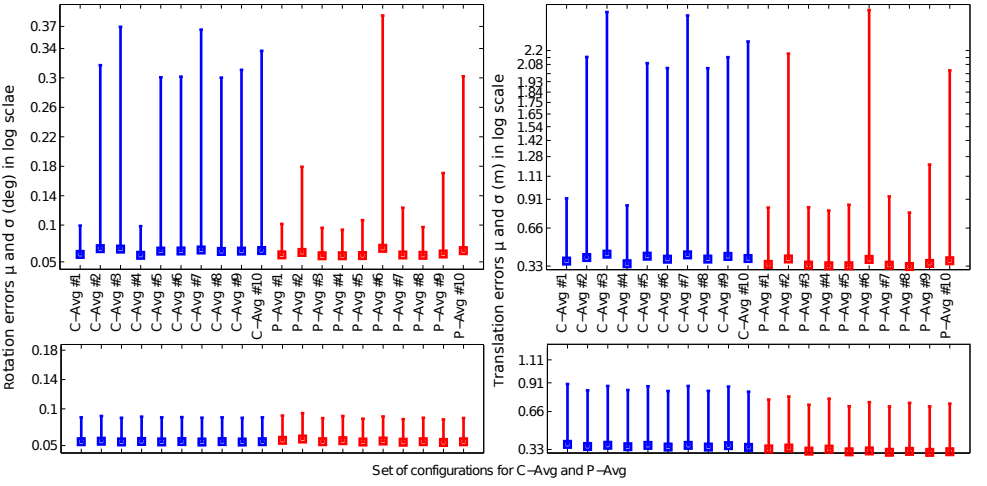

Figure 4: Mean error (box) and std (line) for the tested configurations. Rotation errors (left) and translation errors (right), considering $\approx$ 300 matches in each pair of frames (top) and 2000 matches (bottom).

the three matches for generating the h-th model $\mathcal{X}_s^{(h)} = \{\mathcal{X}_{s,k_1}, \mathcal{X}_{s,k_2}, \mathcal{X}_{s,k_3}\}$ have to be drawn from an uniform distribution with logarithmic increasing boundaries, i.e., $\{k_1, k_2, k_3\}^{(h)} \sim \mathcal{U}(0, c\ log(c\ h))$, with $c = 4$ in our tests. This forces the drawing method to start selecting more candidates from the top of the list (i.e., best scores) until the boundaries are expanded. We will refer to this strategy as the progressive policy, in contrast with the coarse policy.

5 Experimental Results

Here we evaluate the behaviour of the presented method with both polices; the standard Coarse-averaging (C-Avg) and the Progressive-averaging (P-Avg) variation. For this purpose, we use the KITTI Vision Benchmark Suite, which includes challenging sequences of driving scenarios (urban and highways). All the experiments were carried out in an Intel i7-3820 PC at 3.6 GHz, with a single thread. We start by testing the influence of the most relevant parameters of our method; i.e., the number of generated models, the number of models used for averaging and the volume of input data. To evaluate these parameters we defined ten configurations, listed in Table 1. Additionally, we created two different sets of associations for the KITTI sequence 01, with an average amount of 300 and 2000 matches per frame, respectively. The results of this experiment are shown in Fig. 4 according to the mean error and the standard deviation of each configuration. Error measurements are split up into rotation and translation errors for a better understanding of the results. The experiment shows that using more data increases the quality of the results notably, a known fact also studied in [28]. It can also be observed that P-Avg configurations are usually better than their counterparts C-Avg.

Table 1: Parameters for the reference configurations

Configurations	**#1**	**#2**	**#3**	**#4**	**#5**	**#6**	**#7**	**#8**	**#9**	**#10**
# of generated models	100	100	200	200	500	500	1000	1000	2000	2000
# of averaged models	25	50	50	100	125	250	250	500	500	1000

Our second experiment measures the quality of the models obtained from the ℓ_1-averaging process. For that we compare the number of inliers of our models with respect to the inliers

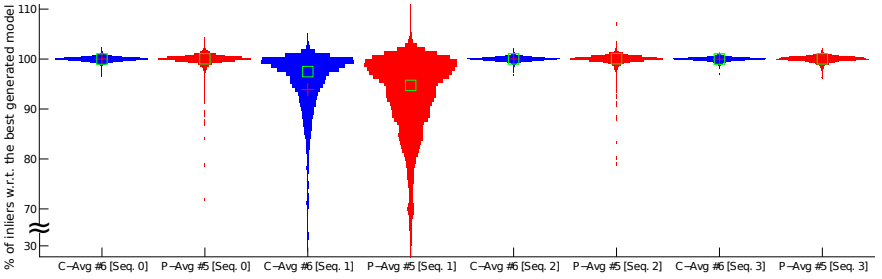

Figure 5: Inliers distributions for C-Avg #6 (blue) and P-Avg #5 (red) in KITTI seq. $00-03$.

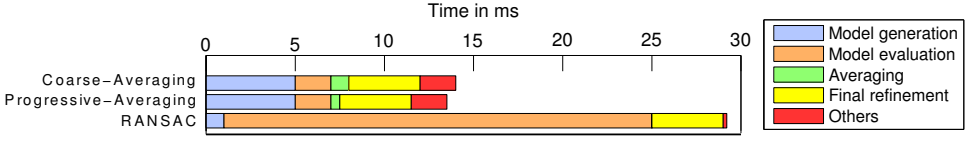

Figure 6: Time diagram comparing the different stages of the approaches under evaluation.

supporting RANSAC best model (RANSAC is limited to 100 hypotheses for real-time purposes). This test is carried out with configurations C-Avg #6 and P-Avg #5, as they offer a good trade-off between efficiency and accuracy. Tests are performed in the KITTI sequences 00-03. The results are depicted in Fig. 5 as violin charts. It can be observed that the proportion of inliers in our models is usually the same as the one in the RANSAC model. Results for sequence 01 tend to be worse due to the special conditions of the scene— a highway with repetitive textures. Notice also that in some situations our models present more inliers than the RANSAC model, which is due to the "extra" local optimization stage naturally inherent to the averaging process.

The third experiment presents a comparison between the proposed approach, RANSAC and a least-squares unrobustified version of Eq. 2. The test consists in performing *VO* for the KITTI sequences 00 and 02 with an average amount of 2000 correspondences per pair of frames. The configurations tested are C-Avg #6 and P-Avg #5, while RANSAC is configured to generate and evaluate just up to 100 hypotheses in order to meet the real-time constraint. The kind of error considered here is at the level of individual poses, in other words, we measure the error for each pose of the vehicle with respect to the ground truth trajectory and provide the mean and the standard deviation for the entire sequence. Fig. 1 shows the trajectory estimation for all the methods along with the ground truth. Both, C-Avg and P-Avg, reach the same level of accuracy as RANSAC. In sequence 00 the average error per pose of all the robust method is around 4 cm for the translation and 0.07 deg for the rotation. Similar values are obtained for the sequence 02. It is important to notice that, although the results are very similar, the averaging strategy takes considerably less time. Fig. 6 presents a time summary of the three robust methods for each of their relevant stages. The average time consumed for C-Avg and P-Avg is half of the required by RANSAC. The coarse evaluation only takes 2 ms to score 500 models and the ℓ_1-averaging is performed in less than 1 ms for 250 models. The triangulation of the points and the generation of $\boldsymbol{M_l}$ and $\boldsymbol{M_r}$ takes 2 ms. The common stage of final refinement with the entire set of inliers takes an average of 4 ms for all the techniques. Additionally, some preliminary experiments have been carried out to compare our approach with PROSAC. At first glance it seems that PROSAC achieves slightly more accurate results than RANSAC, as observed with C-Avg and P-Avg, but maintaining similar execution times. Further analysis is still required to clarify this issue.

6 Conclusion and Future Work

We have presented a novel technique for the problem of robust camera-pose estimation with application to *VO-VSLAM* frameworks in large driving scenarios. The approach is based on the combination of a very fast coarse-evaluation function and a robust ℓ_1-averaging procedure. This scheme is more suitable for dealing with large amount of data, which as we showed, helps producing very accurate results. Our experiments in real driving scenarios showed that the proposed approach produces same quality of results as RANSAC while taking considerably less time.

As future work, we consider interesting to investigate the use of this kind of approaches in combination with structure-less global optimization methods such as [19][24], which can benefit from our fast camera-pose estimation as an initialization stage. Furthermore, it seems that a large amount of the computation carried out by our method can be directly reused by these techniques, hopefully leading to very efficient solutions.

References

[1] S. Choi, T. Kim, and W. Yu. Performance evaluation of RANSAC family. In *Proceedings of the BMVC*, pages 81.1–81.12, London, UK, 2009. BMVA Press.

[2] O. Chum and J. Matas. Matching with PROSAC – progressive sample consensus. In *Proceedings of the IEEE CVPR*, pages 220–226, Washington, DC, USA, 2005.

[3] A.I. Comport, E. Malis, and P. Rives. Accurate quadrifocal tracking for robust 3D visual odometry. In *Proceedings of the IEEE ICRA*, Rome, Italy, 2007.

[4] A.I. Comport, E. Malis, and P. Rives. Real-time quadrifocal visual odometry. *International J. of Robotics Research, Special issue on Robot Vision*, 29(2-3):245–266, 2010.

[5] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE TPAMI*, 2007.

[6] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6), 1981.

[7] J. Gallier. Notes on differential geometry and Lie groups. Technical report, University of Pennsylvania, Department of Computer and Information Science, 2012.

[8] M. Garrigues and A. Manzanera. Real time semi-dense point tracking. In *Proceedings of the ICIAR*, pages 245–252, Berlin, Heidelberg, 2012.

[9] A. Geiger, M. Roser, and R. Urtasun. Efficient large-scale stereo matching. In *Proceedings of the ACCV - Volume Part I*, pages 25–38, Berlin, Heidelberg, 2011.

[10] A. Geiger, J. Ziegler, and C. Stiller. Stereoscan: Dense 3D reconstruction in real-time. In *Proceedings of the IEEE IV Symposium*, pages 963–968, Baden-Baden, Germany, 2011.

[11] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Proceedings of the IEEE CVPR*, Providence, USA, 2012.

[12] V. M. Govindu. Lie-algebraic averaging for globally consistent motion estimation. In *Proceedings of the IEEE CVPR*, Washington, DC, USA, 2004.

[13] V. M. Govindu. Robustness in motion averaging. In *Proceedings of the ACCV*, Berlin, Heidelberg, 2006. Springer-Verlag.

[14] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. *Robust Statistics: The Approach Based on Influence Functions (Wiley Series in Probability and Statistics)*. Wiley-Interscience, New York, 2005.

[15] R. Hartley. Minimizing algebraic error in geometric estimation problems. In *Proceedings of the IEEE ICCV*, Washington, DC, USA, 1998.

[16] R. Hartley, K. Aftab, and J. Trumpf. L1 rotation averaging using the Weiszfeld algorithm. In *Proceedings of the IEEE CVPR*, Washington, DC, USA, 2011.

[17] R. Hartley, J. Trumpf, Y. Dai, and H. Li. Rotation averaging. *International J. of Computer Vision*, pages 1–39, 2013.

[18] P. J. Huber. *Robust Statistics*. Wiley Series in Probability and Statistics - Applied Probability and Statistics Section Series. Wiley, 2004.

[19] V. Indelman, R. Roberts, C. Beall, and F. Dellaert. Incremental light bundle adjustment. In *Proceedings of the BMVC*, Guildford, UK, 2012.

[20] B. Kitt, A. Geiger, and H. Lategahn. Visual odometry based on stereo image sequences with RANSAC-based outlier rejection scheme. In *Proceedings of the IEEE IV Symposium*, San Diego, CA, USA, 2010.

[21] D. Nistér. Preemptive RANSAC for live structure and motion estimation. In *Proceedings of the IEEE ICCV*, pages 199–, Washington, DC, USA, 2003.

[22] D. Nistér, O. Naroditsky, and J. Bergen. Visual odometry. In *Proceedings of the IEEE CVPR*, volume 1, Washington, DC, USA, 2004.

[23] R. Raguram, J. Frahm, and M. Pollefeys. A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus. In *Proceedings of the ECCV*, pages 500–513, Berlin, Heidelberg, 2008. Springer-Verlag.

[24] A. Rodríguez, P. López-de-Teruel, and A. Ruiz. Reduced epipolar cost for accelerated incremental SfM. In *Proceedings of the IEEE CVPR*, Colorado Springs, USA, 2011.

[25] G. Ros, J. Guerrero, A. D. Sappa, D. Ponsa, and A. M. López. VSLAM pose initialization via Lie groups and Lie algebras optimization. In *Proceedings of the IEEE ICRA*, Karlsruhe, Germany, 2013.

[26] G. Sibley, C. Mei, I. Reid, and P. Newman. Vast-scale outdoor navigation using adaptive relative bundle adjustment. *International J. of Robotics Research*, 2010.

[27] H. Strasdat, A. J. Davison, J. M. M. Montiel, and K. Konolige. Double window optimisation for constant time visual SLAM. In *Proceedings of the IEEE ICCV*, Barcelona, Spain, 2011.

[28] H. Strasdat, J. M. M. Montiel, and A. J. Davison. Visual SLAM: Why filter? *Image Vision Comput.*, 30(2):65–77, February 2012.