SAC-GNC: SAmple Consensus for adaptive Graduated Non-Convexity

Valter Piedade[1], Chitturi Sidhartha[2], José Gaspar[1], Venu Madhav Govindu[2], Pedro Miraldo[3]
[1]Instituto Superior Técnico, Lisboa, Portugal [2]Indian Institute of Science, Bengaluru, India
[3]Mitsubishi Electric Research Labs (MERL), Cambridge, MA, USA

Abstract

Outliers are ubiquitous in geometric vision contexts such as pose estimation and mapping, leading to inaccurate estimates. While robust loss functions can tackle outliers, it is challenging to make the estimation robust to the choice of initialization and to estimate the appropriate robust loss shape parameter that allows distinguishing inliers from outliers. Graduated non-convexity (GNC) often mitigates these issues. However, typical GNC uses a fixed annealing factor to update the shape parameter, which can lead to low-quality or inefficient estimates. This paper proposes a novel approach to adaptively anneal the shape parameter within a GNC framework. We developed a search strategy that incorporates a sampling of annealing choices and model scorings to select the most promising shape parameter at each GNC iteration. Additionally, we propose new stopping criteria and an initialization technique that improves performance for diverse data, and we show the benefits of combining discrete and continuous robust estimation strategies. We evaluate our method using synthetic and real-world data in two problems: 3D registration and pose graph optimization in SLAM sequences. Our results demonstrate greater efficiency and robustness compared to previous GNC schemes. Code and other resources are available at `https://www.merl.com/research/highlights/sac-gnc`.

1. Introduction

Least squares estimation is a primary method for solving non-linear problems in computer vision. The goal is to find the best model parameters θ^* such that

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{N} r^2\left(\boldsymbol{x}_i, \theta\right), \tag{1}$$

where $r\left(\boldsymbol{x}, \theta\right)$ is a function that outputs the residuals given input data $\boldsymbol{x}$ and model θ. However, least squares solvers are extremely sensitive to outliers since the square of the residual increases drastically for outlier observations and thus significantly affects the final estimate. This is especially critical in computer vision tasks such as pose estimation and pose graph optimization, where outlier correspondences between images/scans are inevitable due to repeated structures, inaccurate feature detection, matching, *etc*.

An alternative to least squares estimation is M-estimation, which utilizes a robust loss function $\rho_\sigma\left(\cdot\right)$ to mitigate the influence of outliers in the estimation process. M-estimation changes Eq. **1** to

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{N} \rho_\sigma\left(r\left(\boldsymbol{x}_i, \theta\right)\right). \tag{2}$$

Although there have been many alternatives for $\rho_\sigma(\cdot)$, lately, many researchers have been using the Geman-McClure robust loss [7, 52, 60], defined as

$$\rho_\sigma\left(r\left(\boldsymbol{x}_i, \theta\right)\right) \stackrel{\text{def}}{=} \frac{r^2(\boldsymbol{x}_i,\theta)}{1+\frac{r^2(\boldsymbol{x}_i,\theta)}{\sigma^2}}, \tag{3}$$

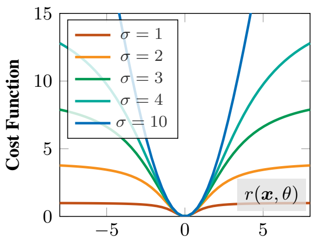
where σ is a hyperparameter that characterizes the shape of the function and is thus referred to as the shape parameter. Figure **1** displays the Geman-McClure loss for varying σ. This work primarily uses the Geman-McClure, but we also experiment with Cauchy, Bisquare, and Logistic losses [19].

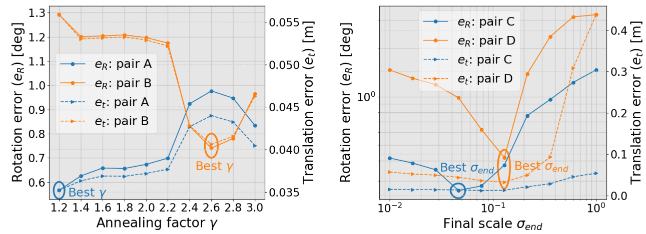
Graduated Non-Convexity & Limitations: For an ideal σ that accurately captures the noise level in the data, $\rho_\sigma(.)$ can turn Eq. **2** robust to outliers[1]. However, even for a predefined ideal σ, which might be impossible to set in general scenarios, it is unlikely that we will obtain an accurate solution. This happens because, for any ideal σ, Eq. **2** is highly non-convex, hence making any local optimization routine susceptible to the choice of the initial guess for θ. A commonly used technique to mitigate this problem is Graduated Non-Convexity (GNC) [8, 43].

GNC starts with a large value for the shape parameter σ, where the loss function is convex, approximating the estimation of least squares. Then, by iteratively reducing σ and

[1]For an ideal σ, the outliers will lie on the saturation part of the robust loss function as shown in Fig. **1**, and therefore have null gradients in the optimization problem in Eq. **2**.

Figure 1. Illustration of the Geman-McClure cost function $\rho_\sigma(\cdot)$ for varying shape parameter σ values. As σ decreases, the saturation point decreases, as does the overall value of the robust cost function.





(a) Varying annealing factor γ_{GNC} (b) Varying the final shape σ_{end}

Figure 2. Illustration of GNC's drawbacks in a 3D registration problem. Using two pairs of a sequence of the 3DMatch dataset (namely the one in Fig. 3), we solve the standard GNC algorithm (described later in Algorithm 1) varying (a) the annealing factor γ_{GNC}, and (b) the final shape σ_{end}. By varying the annealing factor, we observe that the best γ_{GNC} differs from pair A to pair B and that using the ideal value of one pair on the other leads to poor estimates. Varying σ_{end} leads to similar observations since the best σ_{end} for pairs C and D also differ and would lead to worse results if exchanged. The supplementary material provides a more in-depth analysis of the best values of γ_{GNC} and σ_{end} over a larger data set.

solving Eq. **2**, GNC progressively moves toward the non-convex cost target with σ_{end}. Most *previous GNC-based approaches use a non-optimal fixed annealing factor* γ_{GNC} to update σ (*e.g*., FGR [68], TEASER++ [61], and [60]), which can lead to poor estimates and/or harm efficiency. Some authors have worked on an adaptive annealing factor, namely GNCp [52]. Although efficient in some scenarios, the method in [52] is problem-specific (additional problem-specific computation is required), becomes intractable for high-dimensional θ, and struggles with low inlier rates.

In addition to the annealing factor, it is equally *challenging to set an ideal value for the final shape parameter* σ_{end}, which can vary based on the data (even within the same dataset) since it relates to its noise level. Furthermore, although *most previous approaches assume that the next iteration estimate is always better than the previous*, a non-ideal σ_{end} can lead to unwanted outliers in the estimation or removing inliers from the optimal (unknown) inlier set. Figure **2** illustrates the GNC issues in selecting a fixed γ_{GNC} and σ_{end}. It can be observed that the ideal values for γ_{GNC} and σ_{end} vary with the data. Therefore, no general value for either parameter can be predefined and consistently lead to the most accurate estimate.

Lastly, *previous GNC-based methods improve model estimation by assuming a continuous decrease of* σ. Although this makes sense to increase the non-convexity of the optimization problem for fixed annealing updates, for adaptive updates, as in [52], this has the strong limitation of not being able to recover from a poorly chosen annealing parameter at earlier stages of GNC.

Our Contributions: To address the mentioned limitations of GNC, we propose a new adaptive annealing strategy for GNC (see Algorithm **2**). To deal with the limitations of having a fixed annealing factor, we sample various annealing factors at each iteration and decide which shape parameter to follow in the next by leveraging model scorings. To mitigate issues related to continuously decreasing σ, our method uses a search-like strategy to choose the best model to test at each GNC iteration. To conclude, our algorithm is the first kind of GNC to include a stopping criteria based on model scoring, which significantly alleviates the hard choice of σ_{end}. We call this approach SAC-GNC. In addition to the previous contributions, we propose an initialization technique that reduces reliance on predefined thresholds and improves efficiency. We evaluate SAC-GNC in 3D registration and pose graph optimization problems, which are core problems of SLAM [14] and SfM [41, 50] pipelines and are typically solved using GNC-based methods. To sum up:

1. We propose a new algorithm for robust and efficient estimation using a GNC-type approach that utilizes an adaptive annealing strategy based on sample and consensus;
2. While some researchers see GNC-like and sample & consensus methods as contrasting approaches (continuous *vs*. discrete estimators) with different benefits, in this paper, we show that combining sample & consensus into GNC has benefits over previous GNC approaches;
3. Experiments show that our algorithm outperforms baselines in accuracy and efficiency.

We developed a C++ framework with Python wrappers to support testing of Graduated Non-Convexity approaches.

2. Related Work

2.1. Robust optimization estimation

It is known that non-minimal solvers such as [13, 56, 67] struggle with outliers. The state-of-the-art approaches are based on iterative solvers, which require a good initial guess to ensure a good model estimation. Several techniques such as branch and bound (BnB) [30, 40], semidefinite programming (SDP), and sums of squares (SOS) relaxations [9] have been developed to help in the convergence. Both SDP [12, 17, 18] and SOS [37, 45] have been successfully used to develop solutions with optimality guarantees. [55] proposes a general framework for robust estimation, where each iteration progressively increases the proportion of outliers filtered out of the estimate. Other methods were specifically derived to deal with outliers:

M-estimators: M-estimators are a class of robust solvers that minimize robust cost functions, as denoted in Eq. **2**. The Geman-McClure loss is among the most utilized robust functions in recent years. Cauchy, Bisquare, Logistic, and others (see [19]) are also used in M-estimation. [36] compares the performance of various robust loss functions.

Graduated non-convexity: GNC is a popular estimation algorithm [8, 43, 60, 62, 64]. While M-estimation does not change the value of σ throughout the estimation, GNC approaches iteratively adjust its value according to an annealing factor γ_{GNC}. GNC has been particularly used in 3D registration [60, 61, 68] and pose graph optimization [60].

Although various authors have developed GNC-based algorithms, little attention has been paid to defining the annealing factor and the stopping criterion (*i.e.*, σ_{end}). As shown in Fig. **2**, the ideal values for these parameters depend on the data. However, previous approaches such as FGR [68], TEASER++ [61], and [60] use fixed values for γ_{GNC} and σ_{end}. This prevents these approaches from obtaining the most accurate and/or efficient solutions. Recently, GNCp [52] proposed an adaptive annealing strategy for 3D registration, where the annealing is computed at each iteration from the Hessian of the problem cost function. The authors show that adaptive annealing results in more accurate solutions and requires fewer iterations, although they still use a fixed σ_{end}. In this paper, we also use an adaptive annealing factor. However, we aim for a general solution independent of the underlying problem, unlike GNCp's approach. We propose an algorithm that finds the most suitable annealing factor using a search-like scheme and a model scoring function. Regarding the stopping criterion, we propose new criteria that stop the algorithm when we converge to a solution or all promising σ's have been tested.

2.2. RANSAC-based estimators

RANSAC [28] is an alternative robust estimator that iteratively samples minimal data, estimates a model, and scores it using inlier counting. Its output is the solution with the highest consensus. Despite yielding favorable results for properly tuned parameters, it has some drawbacks. Firstly, RANSAC needs a minimal solver for each problem (e.g., [26, 31, 33, 39]). Secondly, there are issues related to computational complexity. Depending on the problem (*i.e.*, size of the sampling set) and data (*i.e.*, inlier rate), one may need to run RANSAC for numerous iterations to ensure an accurate solution. Several improvements in RANSAC have been proposed, e.g., [1–4, 11, 11, 22, 23, 42, 58].

2.3. Learning-based methods

Learning-based methods have become popular in recent years. A significant improvement was noticed in problems like high-quality feature detection [21, 25, 44, 49, 53] and finding correspondences [20, 35, 48, 51, 53, 63], providing data with lower proportion of outliers. Other methods focus on detecting keypoints, finding correspondences, and estimating model parameters in an end-to-end manner [5, 10, 32, 34, 57, 66]. As our focus is on improving GNC's limitations, we won't compare our strategy with learning-based solutions since these are not GNC-like approaches.

Algorithm 1: *Graduated Non-Convexity*

Input – Let $\mathcal{D}$ be some data; σ_0 and σ_{end} be initial and final shape parameter; and γ_{GNC} be the annealing factor.

Output – Final model θ^*

```
1 Initialize: k ← 1;
2 θ_0 ← computeInitialModel (D);
3 while σ_k ≥ σ_end do
4     σ_k ← updateShape (σ_{k−1}, γ_GNC);            ▷ Eq. 4
5     θ_k ← computeModel (D, θ_{k−1}, σ_k);          ▷ Eq. 2
6     k ← k + 1;
7 θ* ← θ_k;
```

3. Graduated Non-Convexity (GNC)

As described in Sec. **1**, GNC mitigates the risk of local optimization routines converging to poor solutions. This challenge arises because the cost in Eq. **2** becomes highly non-convex for the chosen ideal σ_{end}. To address this, GNC starts with a large σ, which makes $\rho_\sigma(\cdot)$ convex, making Eq. **2** easier to optimize. Then, it progressively reduces σ and solves for θ using the solution at the previous iteration as the initial estimate. This is repeated until σ reaches the predefined value σ_{end}. At that point, the latest estimated model θ_k is returned. This process ensures that after every σ update, we have a good initialization to minimize the corresponding robust cost, which (when working correctly) leads to convergence to a high-quality solution when σ_{end} is reached. Algorithm **1** outlines the standard GNC steps. At each iteration k, σ_k is updated (Line **4**) according to

$$\sigma_k = \frac{\sigma_{k-1}}{\gamma_{\text{GNC}}}, \tag{4}$$

where γ_{GNC} is a fixed annealing factor. Line **5** corresponds to solving Eq. **2** using the updated shape parameter σ_k and previous model estimate θ_{k-1}. Most previous approaches (*e.g.*, [60, 61, 68]) set the annealing factor to 1.4.

Computing the model in Line **5** can be challenging and computationally expensive. Therefore, Line **5** is typically accomplished by following the Black-Rangarajan duality [7] and alternating minimization methods. Additional details are provided in the supplementary material.

4. SAmple Consensus for adaptive GNC

This section proposes a new method for adaptive annealing in GNC entitled SAC-GNC, outlined in Algorithm **2**.

4.1. Method overview

SAC-GNC comprises an annealing sample consensus approach within an *online searching strategy* [46] to find the best shape parameter σ at each GNC iteration. For readability and replicability purposes, we focus on algorithm details for solving the problem instead of modeling the search. In the supplementary material, we show an illustration of the proposed tree-search strategy.

Similar to Algorithm **1**, we start by computing a least squares solution, since no initial model is given. This is shown in Line **2** of Algorithm **2**. To improve the efficiency of the estimation, our first contribution is the definition of initial shape parameter σ_0 based on the initial model residuals at Line **3** (details in Sec. **4.4**). Then, in each iteration k of the GNC cycle, we run several annealing trials (namely T). For each trial t, an annealing factor $\gamma_{k,t}$ is chosen within a given interval defined by $[\gamma_{\text{GNC}} \cdot \alpha^-,\ \gamma_{\text{GNC}} \cdot \alpha^+]$, where $\alpha^\pm$ represents a relaxation of the original annealing parameter γ_{GNC}. We set $\alpha^- = 1$ and $\alpha^+ = 3.5$. Details on the annealing factor selection process are given in Sec. **4.2**. Each $\gamma_{k,t}$ sets a new hypothesis $\mathcal{H}_{k,t}$ comprised of the shape parameter $\sigma_{k,t}$ computed using Eq. **4**, model $\theta_{k,t}$, model score $s_{k,t}$, and level in the search tree $d_{k,t}$ (Lines **10** to **15**). As we are deciding online what is the best σ_k to explore, one might choose a $\sigma_{k,t}$ based on some cost that leads to non-optimal estimations in future iterations. To alleviate this issue, instead of having a Markovian approach of forgetting all hypotheses other than the best one at each iteration, all promising hypotheses are added to a priority queue $\mathcal{Q}$ in Line **16**. In Line **17**, we keep track of the best overall hypothesis–different than standard GNC, we cannot output one of the latest estimated hypotheses since it may be worse than a previous hypothesis due to the search strategy we utilize. Next, we check whether the search is complete or a consensus was reached regarding the quality of the estimate by checking the stopping criteria in Line **19**. Finally, we select the next hypothesis for exploration from the top of the priority queue (most promising one) in Line **20**. See the illustrative example in the supplementary material.

Depending on the number of trials T, the computational complexity of each iteration will increase proportionally. To be efficient, and since each trial is independent of the others, each annealing hypothesis can be run in parallel. In the supplementary material, we show how the number of trials impacts the accuracy and efficiency of the estimation.

SAC-GNC scales for any problem with N variables (*i.e.*, problem size)[2] just like vanilla GNC. SAC-GNC's overhead lies in the scoring mechanism, which depends linearly on the data size M, *i.e.*, $O(M)$ complexity. Preemptive scoring can reduce the scoring complexity, although SAC-GNC has proved to be efficient even for large data.

[2]Unlike GNCp, which sequentially computes Hessian's minimum eigenvalue multiple times per GNC iteration with $O(N^3)$.

Algorithm 2: *SAC-GNC: SAmple Consensus for adaptive GNC*

Input – Let $\mathcal{D}$ be some data; annealing parameter γ_{GNC}; T be the number of trial hypotheses for the annealing factor. Underline means new in this paper.

Output – Best hypothesis $\mathcal{H}^* = \{\sigma^*, \theta^*, s^*, d^*\}$.

1 **Initialize:** $k \leftarrow 0$;
2 $\theta_0 \leftarrow$ computeInitialModel $(\mathcal{D})$;
3 $\sigma_0 \leftarrow$ shapeInitialization (θ_0); ▷ Sec. **4.4**
4 $Q \leftarrow$ empty queue;
5 $\mathcal{H}_0 \leftarrow \{\sigma_0,\ \theta_0,\ \infty, 0\}$;
6 **while** *True* **do**
7     $k \leftarrow k + 1$;
8     $\{\sigma_{k-1},\ \theta_{k-1},\ s_{k-1}, d_{k-1}\} \leftarrow \mathcal{H}_{k-1}$;
9   **for** $t = 1 : T$ **do**
10         $\gamma_{k,t} \leftarrow$ getAnnealing $(\gamma_{\text{GNC}}, \alpha^\pm)$; ▷ Sec. **4.2**
11         $\sigma_{k,t} \leftarrow$ updateShape $(\sigma_{k-1},\ \gamma_{k,t})$; ▷ Eq. **4**
12         $\theta_{k,t} \leftarrow$ computeModel $(\mathcal{D}\ ,\ \theta_{k-1},\ \sigma_{k,t})$; ▷ Eq. **2**
13         $s_{k,t} \leftarrow$ computeScore $(\mathcal{D},\ \theta_{k,t})$; ▷ Sec. **4.2**
14         $d_{k,t} \leftarrow d_{k-1} + 1$;
15         $\mathcal{H}_{k,t} \leftarrow \{\sigma_{k,t},\ \theta_{k,t},\ s_{k,t}, d_{k,t}\}$;
16     $\mathcal{Q} \leftarrow$ addToQueue$(\{\mathcal{H}_{k,i}\})$; ▷ Sec. **4.2**
17     $\mathcal{H}^* \leftarrow$ saveBestHypothesis$(\mathcal{H}^*,\ \{\mathcal{H}_{k,i}\})$; ▷ Sec. **4.2**
18   **if** *stoppingCriteria*$(\mathcal{Q},\ \mathcal{H}^*)$ **then**
19   break; ▷ Sec. **4.3**
20     $\mathcal{H}_k \leftarrow$ getNextHypothesis$(\mathcal{Q})$; ▷ Sec. **4.2**

4.2. Online search for σ

The main contribution of this paper is the search-like estimation of σ, which consists of sampling annealing factors, model scoring (finding consensus), and the definition of a search queue. Each block is described below.

Annealing sampling: Our sample consensus strategy requires sampling T distinct annealing factors at each iteration k. Consider $t = 1, \dots, T$ generated hypothesis for the annealing factor, denoted as $\gamma_{k,t}$. For each t, a shape parameter $\sigma_{k,t}$ is computed according to $\sigma_{k,t} = \sigma_{k-1}/\gamma_{k,t}$, following Eq. **4**. Each $\gamma_{k,t}$ is chosen randomly from a predefined interval such that $\gamma_{k,t} \in [\gamma_{\text{GNC}} \cdot \alpha^-,\ \gamma_{\text{GNC}} \cdot \alpha^+]$[3]. Increasing the sampling size T increases the chances of sampling a reasonable set of annealing factors, albeit at a cost in computational complexity. An alternative sampling process is tested in the supplementary material.

Model scoring: While common GNC approaches consider the current estimate to be better than all previous ones, this is not always true. For example, when σ decreases too much, the gradient of some inliers in Eq. **3** might converge

[3]Notice that we want $\gamma_{k,t} > 1$ to ensure we move towards increasing the non-convexity of the robust cost, which is true for $\gamma_{\text{GNC}} > 1$ because we set $\alpha^- = 1$.

to zero, meaning that those inliers will be neglected in Eq. **2**. To determine the quality of a model, we compute a score $s_{k,t}$ for each hypothesis $\mathcal{H}_{k,t} = \{\sigma_{k,t}, \theta_{k,t}, s_{k,t}, d_{k,t}\}$ obtained by the sampled annealing factors $\gamma_{k,t}$. A trivial scoring could be the weighted or non-weighted model residuals' mean (or median). However, a non-weighted approach would not be robust to outliers, and a weighted approach would mainly give better scores to models obtained from lower $\sigma_{k,t}$ since the Geman-McClure loss is a monotonically increasing function of σ.

Scoring is a critical step in RANSAC-based techniques, which are known to be robust. Therefore, we borrowed some of its typical scoring functions, namely MSAC [54], where $s_{k,t}$ is the sum of the truncated squared residuals. We also test RANSAC [28] (inlier counting) and LMeds [38] (median of squared residuals) scores. MSAC and RANSAC require an inlier threshold. However, this is common for robust estimators in computer vision and is appropriate to use. Results with these scoring functions are given in the supplementary material. Threshold marginalization approaches (*e.g*., MAGSAC [3]) can also be explored.

Priority search queue: Our search strategy employs a priority queue to hold promising hypotheses and decide which to explore next. Each hypothesis can be seen as a node in a tree (from the root level 0 to ∞), where each iteration grows a single node using a branching factor of T, creating new nodes on the next level. The priority queue is sorted primarily using the tree level (depth of the node in the tree), denoted as $d_{k,t}$, which is the number of times σ is slashed, and is stored in $\mathcal{H}_{k,t}$. Secondly, it is sorted by the model score. This priority promotes searching all hypotheses in a tree level before moving to the next (prioritizes a breadth-first search approach, [46]) and exploring the best-scoring models within a tree level first. We note that primarily prioritizing scores would lead to exploring mostly hypotheses from higher tree levels, *i.e*., higher $d_{k,t}$ (would lead to a depth-first search kind of approach, [46]), since a lower σ generally gets better scores, even for inaccurate models, lowering chances of getting an accurate estimate.

For efficiency purposes, some additional heuristics are considered. We do not append all hypotheses $\{\mathcal{H}_{k,i}\}, i = 1, \cdots, T$ to the priority queue since it would increase the computational time exponentially. For the same reason, we set a maximum size for the priority queue, denoted Q_{size}. We specify the maximum number of new hypotheses added in each iteration using Q_{add}. At each GNC iteration k, we decide which hypotheses to add to the queue as follows:

1. Hypotheses with a σ below a preset σ_{min} are not added;
2. The best scoring hypothesis is always added (except if not complying with item **1**);
3. No more than Q_{add} new hypotheses are added, and;
4. Only hypotheses with a similar score but a sufficiently different model[4] are added.

After adding hypotheses, if the queue size surpasses Q_{size}, the hypotheses with lower priority are discarded. Increasing the Q_{add} and Q_{size} will allow for a more thorough search, albeit at a computational cost. Supplementary material shows an ablation study for Q_{add} and Q_{size}.

Best hypothesis: Given the search nature of our algorithm, we cannot retain the best of the last estimated hypotheses $\{\mathcal{H}_{k,i}\}, i = 1, \cdots, T$ as the best hypothesis overall ($\mathcal{H}^*$) because we may end up testing some shape that results in a poorer estimate. Thus, in each iteration, we check if any of the new hypotheses $\{\mathcal{H}_{k,i}\}$ has a better score $s_{k,i}$ than the score s^* of the best hypothesis $\mathcal{H}^*$ (Line **17**). When the stopping criteria are triggered, $\mathcal{H}^*$ is returned.

4.3. Stopping criteria

While GNC-based methods typically iterate until a certain shape σ_{end} is reached, we propose removing this termination criterion because it is not robust to diverse data. Instead, we follow typical optimization techniques of checking for model or scoring convergence. If in consecutive search tree levels, the model or the model scoring does not differ by more than a predefined threshold, we end the estimation. In addition, when the queue is empty (*i.e*., there are no more hypotheses to explore), the estimation is also stopped. While we still predefine a σ_{min} value (item **1** of Sec. **4.2**) to add new hypotheses, we set it to a much lower value than the typical σ_{end} of GNC in Algorithm **1**, which makes SAC-GNC stop mainly from the convergence criterion. We choose σ_{min} only to ensure that Eq. **2** does not vanish. Unlike σ_{end}, σ_{min} does not depend on the noise of the data or the outliers. We address this in the supplementary material.

4.4. Shape parameter initialization

GNC starts with a model θ_0 obtained from a least squares solution (Eq. **1**), which does not rely on σ. For the Geman-McClure specifically, this means that σ_0 takes a large enough value. In practice, setting σ_0 too high can lead to successive iterations with no changes to the best model estimate since all data will be considered inliers. To improve efficiency, we offer the option to set σ_0 to the lowest value of σ that approximates the least squares estimation. We compute the residuals $\mathcal{R}_0$ of the initial model, take the maximum value $r_{\text{max}} \in \mathcal{R}_0$, and compute σ_0 by analytically modeling the contribution of the residual to the optimization in Eq. **2**, which we denote as $w \in (0, 1]$ (due to space limitation, we show how w is modeled in the supplementary material). Specifically, we want a σ_0 that makes the w of the data point with maximum residual, r_{max}, close to 1. The initial shape σ_0 is then given by

$$w = \left(\frac{1}{1+r_{\text{max}}^2/\sigma_0^2}\right)^2 \Rightarrow \sigma_0 = \frac{r_{\text{max}}}{\sqrt{(w^{-}-1)}}. \quad (5)$$

[4]Comparison against the best scoring hypothesis at iteration k.

In practice, for efficiency, we define $w = 0.95$. This initialization can be adapted to other robust losses and used by any GNC-based approach.

An alternative to our initialization is [60], where $\sigma_0^2 = 2r_{\text{max}}^2/c^2$ and c is a fixed scale parameter. In the supplementary material, we compare both initialization approaches.

5. Experiments and Results

We evaluate our method on the two computer vision problems that are typically solved utilizing GNC-based approaches: 1) Section **5.1** presents ablation studies and evaluations against baseline methods for 3D registration; 2) Section **5.2** evaluates SAC-GNC in the pose graph optimization (PGO). Additional ablation studies and results are provided in the supplementary material.

Evaluation metrics: Both problems estimate rotations and translations in 2D or 3D. For the rotation error, we use

$$e_R\left(\widehat{R}, R_{\text{gt}}\right) = \text{arcos}\left(\left(\text{trace}\left(\widehat{R}^T R_{\text{gt}}\right) - 1\right)/\eta\right) \quad (6)$$

where $\widehat{R}$ and R_{gt} are the estimated and ground truth rotations, respectively, and η is either 1 or 2 for 2D or 3D, respectively. The translation error is

$$e_{\boldsymbol{t}}\left(\widehat{\boldsymbol{t}}, \boldsymbol{t}_{\text{gt}}\right) = \|\widehat{\boldsymbol{t}} - \boldsymbol{t}_{\text{gt}}\|_2^2, \quad (7)$$

where $\widehat{\boldsymbol{t}}$ and $\boldsymbol{t}_{\text{gt}}$[5] are the estimated and ground truth translations. Similar to recent works [6, 63], in the tables and graphs, we show the mean Average Accuracy (mAA) for each error metric. This robust error function measures the area under the curve of the cumulative distribution of the errors up to a certain threshold (higher mAA is better). We use the average of all the runs for time and iterations.

Method settings: The settings used are $\sigma_{\text{min}} = 10^{-3}$, $\alpha^- = 1$, $\alpha^+ = 3.5$, our σ initialization, random annealing selection, and MSAC [54] scoring. For a fair comparison, we use $\gamma_{\text{GNC}} = 1.4$, as all the baselines do. When not mentioned otherwise, we use Geman-McClure as $\rho_\sigma(\cdot)$. Additionally, we use two versions of our method:

SAC-GNC : $T = 5$, $Q_{\text{add}} = 1$, and $Q_{\text{size}} = 1$;
SAC-GNC++ : $T = 10$, $Q_{\text{add}} = 2$, and $Q_{\text{size}} = 10$.

SAC-GNC prioritizes efficiency, while SAC-GNC++ prioritizes accuracy (narrow *vs.* deep tree search). These settings were ablated in a single 3DMatch [65] sequence for the 3D registration problem and are fixed for all experiments in Sec. **5.1** and Sec. **5.2**. The supplementary material provides more robustness studies for all these parameters.

Item **4** of Sec. **4.2** requires some empirical configurations regarding model similarity. In 3D registration, two models are similar if they differ by less than $5\,\text{deg}$ and 30 cm in rotation and translation, respectively. In PGO, two solutions are similar if the gain between the cost function values is below 1%. Besides, in both problems, only hypotheses with scores that differ by less than 10% compared to the current iteration's best hypothesis can be added. The stopping criteria use the previous settings to check model similarity, and for score convergence, it checks if the best hypothesis $\mathcal{H}^*$ score stopped improving in consecutive tree levels.

[5] Note that $\boldsymbol{t}$ means translation, while t in Algorithm **2** means trial.

5.1. 3D registration problem

Given 3D point correspondences between two point clouds A and B, represented by the tuple $(\boldsymbol{p}_i^A, \boldsymbol{p}_i^B)$, where $\boldsymbol{p}_i \in \mathbb{R}^3$ and $i = 1, \cdots, N$, potentially outlier-contaminated, the goal is to find a rotation $R \in \text{SO}(3)$ and translation $\boldsymbol{t} \in \mathbb{R}^3$ that aligns the two point clouds, *i.e.*,

$$r(\boldsymbol{p}_i^A, \boldsymbol{p}_i^B, R, \boldsymbol{t}) \stackrel{\text{def}}{=} \|\boldsymbol{p}_i^A - R\boldsymbol{p}_i^B - \boldsymbol{t}\|_2, \quad (8)$$

which is plugged into Eq. **2**, where $\boldsymbol{x}_i = (\boldsymbol{p}_i^A, \boldsymbol{p}_i^B)$ and $\theta = (R, \boldsymbol{t})$. Following GNCp [52], we solve Eq. **2** for residuals in Eq. **8** by finding Umeyama's solution [56].

Datasets: Following [52], we evaluate our approach with synthetic and real-world data. For synthetic data, we use ModelNet [59] with Predator [32] features. Results with other synthetic data are provided in the supplementary material. For real-world data, we use 3DMatch [65] (8 sequences) to generate two sets of matches: 1) matches with a lower inlier rate ($\approx 11.6\%$), obtained from FPFH [47] features, and 2) matches with a higher inlier rate ($\approx 59.8\%$), obtained from FCGF [21] features. With KITTI [29], we use matches from FCGF features for the testing sequences 08-10. All features were matched using nearest-neighbor matching. The inlier threshold for model scoring was set to 5 cm in ModelNet and 3DMatch and 20 cm in KITTI. These values resemble the voxel size of the point clouds.

5.1.1. Ablations

Ablation studies use solely the HOME1 sequence of 3DMatch for the low inliers dataset. Below, we provide three ablation studies–more are available in the supplementary material. Each experiment was repeated 10 times.

Study of each component of SAC-GNC: Table **1** weighs different components of SAC-GNC. The first line of the table corresponds to Vanilla GNC. We note that removing σ search means using a fixed γ_{GNC}. In this setting, we cannot test our termination criteria, and we use the same σ_{end} as the termination criterion of vanilla GNC. By default (not using our σ initialization), σ_0 is set to a predefined high number. We observe that using the σ search (even on its own) brings the most improvements, particularly in accuracy. Adding termination (*i.e.*, checking for convergence) or initialization significantly improves the computation time. Without being

σ Search (Sec. 4.2) SAC-GNC	σ Search (Sec. 4.2) SAC-GNC++	Termination (Sec. 4.3)	σ Init. (Sec. 4.4)	mAA ↑ $(e_R, 5°)$	mAA ↑ $(e_t, 0.3m)$	Iter. ↓	Time ↓ [ms]
				0.465	0.612	28	7.45
			✓	0.457	0.605	13.8	5.82
✓				0.487	0.619	11.9	7.47
✓		✓		0.487	0.618	10.9	6.80
✓			✓	0.490	0.622	7.83	6.69
✓		✓	✓	0.490	0.623	6.72	5.98
	✓			0.506	0.644	45.1	58.0
	✓	✓		0.506	0.645	36.4	48.3
	✓		✓	0.507	0.645	39.8	55.7
	✓	✓	✓	0.505	0.644	30.9	46.4

Table 1. Ablation study on the different components of SAC-GNC. All modules disabled correspond to vanilla GNC.

Dataset: 3DMatch	Annealing Update Fixed γ	Annealing Update Adaptive	mAA ↑ $(e_R, 5°)$	mAA ↑ $(e_R, 10°)$	mAA ↑ $(e_t, 0.3m)$	mAA ↑ $(e_t, 0.6m)$	Iter. ↓	Time ↓ [ms]
↑ inliers ≈ 59.8%	1.4	–	0.641	0.798	0.809	0.887	28	5.82
	2.5	–	0.650	0.802	$\underline{0.816}$	$\underline{0.891}$	11	2.60
	3.5	–	0.649	0.799	0.813	0.887	8	$\underline{1.99}$
	5.0	–	0.646	0.799	0.812	0.887	**6**	**1.54**
	–	**SAC-GNC**	$\underline{0.654}$	$\underline{0.805}$	$\underline{0.816}$	0.889	$\underline{6.40}$	3.85
	–	**SAC-GNC++**	**0.655**	**0.810**	**0.819**	**0.896**	11.6	11.2
↓ inliers ≈ 11.6%	1.4	–	0.465	0.610	0.612	0.704	28	7.45
	2.5	–	0.467	0.592	0.594	0.668	11	3.56
	3.5	–	0.463	0.585	0.586	0.660	8	$\underline{2.75}$
	5.0	–	0.446	0.562	0.566	0.639	**6**	**2.21**
	–	**SAC-GNC**	$\underline{0.490}$	$\underline{0.623}$	$\underline{0.623}$	$\underline{0.708}$	$\underline{6.72}$	5.98
	–	**SAC-GNC++**	**0.505**	**0.646**	**0.644**	**0.726**	30.9	46.4

Table 2. Fixed vs. adaptive annealing update. Factors $\gamma = [1.4, 2.5, 3.5, 5.0]$ were chosen arbitrarily since each scan pair has an unknown ideal value. We highlight the **first** and $\underline{\text{second}}$ best.

coupled with σ search, our initialization improves time but lowers slightly the accuracy of vanilla GNC. This is due to our initialization being greedy, and vanilla GNC cannot recover from it as well as an adaptive search.

Fixed vs. adaptive annealing factor: To compare the fixed and adaptive annealing strategies, we replace the multiple trials in Algorithm **2** with a single trial using a fixed predefined annealing factor and compare the results with our algorithm on the HOME1 sequence data with low and high inlier rates. Results are shown in Tab. **2**. We observe a clear trade-off between accuracy and efficiency when varying the fixed annealing factor. We also note that 1) SAC-GNC and SAC-GNC++ have better accuracy than all fixed annealing approaches, and 2) SAC-GNC is more efficient than having the typical 1.4 fixed annealing factor.

Study on the use of different robust cost functions: SAC-GNC applies to any robust loss $\rho_\sigma(\cdot)$, and its choice influences σ initialization (trivial to derive for other $\rho_\sigma(\cdot)$), the model estimation, and σ's update direction. In Tab. **3**, we show results comparing SAC-GNC to vanilla GNC using the Geman-McClure, Cauchy, Bisquare, and Logistic losses. For a fair comparison, we use the same fixed initial σ_0 for all. Comparing the performance of each loss, we observe that the Geman-McClure has the best accuracy-efficiency relation. Comparing the performance of each

Method	Cost Function $\rho_\sigma(\cdot)$	mAA ↑ $(e_R, 5°)$	mAA ↑ $(e_t, 0.3m)$	Iter. ↓	Time ↓ [ms]
Vanilla GNC $\gamma = 1.4$	Geman-McClure	0.465	0.612	28	7.45
	Cauchy	0.425	0.532	35	9.95
	Bisquare	0.147	0.329	22	4.37
	Logistic	0.162	0.269	42	13.2
SAC-GNC	Geman-McClure	0.487	0.618	10.9	6.80
	Cauchy	0.427	0.529	11.2	6.97
	Bisquare	0.276	0.456	5.69	3.26
	Logistic	0.157	0.265	10.5	5.81

Table 3. Ablation study on the use of different robust cost functions. SAC-GNC applies to any robust loss $\rho_\sigma(\cdot)$.

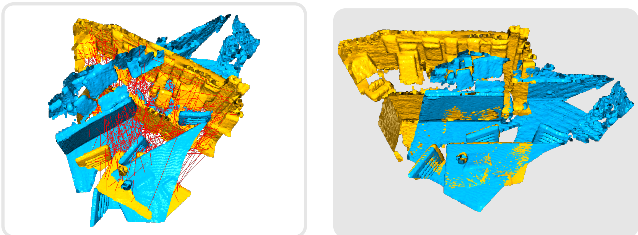
Dataset	Method	mAA ↑ $(e_R, 5°)$	mAA ↑ $(e_R, 10°)$	mAA ↑ $(e_t, 0.3m)$	mAA ↑ $(e_t, 0.6m)$	Time [ms] ↓
ModelNet inliers ≈ 50.8%	RANSAC [28]	0.437	0.671	0.939	0.967	**0.789**
	FGR [68]	0.598	0.752	0.961	0.979	3.05
	TEASER++ [61]	0.613	0.763	0.963	0.980	6.53
	GNCp [52]	0.627	0.771	$\underline{0.967}$	$\underline{0.982}$	$\underline{0.960}$
	SAC-GNC	$\underline{0.708}$	$\underline{0.816}$	**0.976**	**0.987**	1.02
	SAC-GNC++	**0.709**	**0.817**	**0.976**	**0.987**	1.86
3DMatch ↑ inliers ≈ 59.8%	RANSAC [28]	0.492	0.707	0.713	0.829	$\underline{5.13}$
	FGR [68]	0.514	0.693	0.706	0.804	16.5
	TEASER++ [61]	Runtime fail (> 30 minutes per instance)				
	GNCp [52]	0.553	0.726	0.738	0.831	5.61
	SAC-GNC	$\underline{0.584}$	$\underline{0.750}$	$\underline{0.756}$	$\underline{0.842}$	**3.98**
	SAC-GNC++	**0.586**	**0.753**	**0.759**	**0.846**	18.4
3DMatch ↓ inliers ≈ 11.6%	RANSAC [28]	0.279	0.446	0.455	0.577	21.6
	FGR [68]	0.271	0.417	0.443	0.549	$\underline{12.6}$
	TEASER++ [61]	0.287	0.441	0.429	0.555	125
	GNCp [52]	0.348	0.491	0.509	0.603	25.7
	SAC-GNC	$\underline{0.421}$	$\underline{0.539}$	$\underline{0.544}$	$\underline{0.616}$	**7.05**
	SAC-GNC++	**0.435**	**0.560**	**0.565**	**0.640**	56.6
		$(e_R, 1°)$	$(e_R, 5°)$	$(e_t, 0.3m)$	$(e_t, 1m)$	**Time [ms] ↓**
KITTI inliers ≈ 27.4%	RANSAC [28]	0.589	0.888	**0.463**	**0.798**	17.8
	FGR [68]	0.628	0.909	0.298	0.703	62.4
	TEASER++ [61]	0.296	0.634	0.221	0.590	4075
	GNCp [52]	$\underline{0.658}$	0.919	0.346	0.759	**8.39**
	SAC-GNC	**0.693**	$\underline{0.927}$	$\underline{0.363}$	0.769	$\underline{14.3}$
	SAC-GNC++	**0.693**	**0.928**	$\underline{0.363}$	$\underline{0.770}$	66.7

Table 4. 3D registration results on synthetic (ModelNet) and real data (3DMatch and KITTI). We highlight the **first** and $\underline{\text{second}}$ best.

method with each loss, we observe that SAC-GNC outperforms vanilla GNC with all losses except the Logistic. In this case, SAC-GNC takes half the time of vanilla GNC but experiences a slight drop in accuracy. The supplementary material provides this experiment for SAC-GNC++.

5.1.2. Results

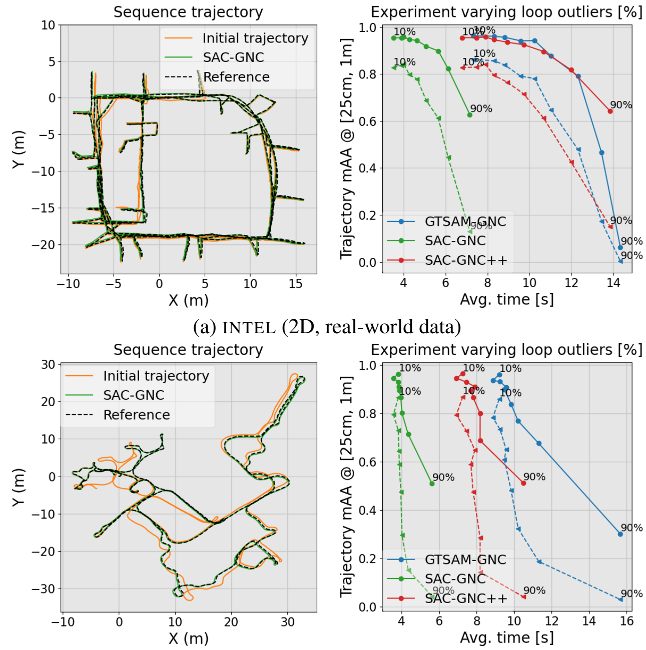
We compare our approach against RANSAC [28], FGR [68], TEASER++ [61], and GNCp [52]. We use the Open3D [69] implementation of RANSAC (with parallel computing), the publicly available codes for FGR and TEASER++, and the original implementation for GNCp, provided by the authors. For a fair comparison, we set the maximum iterations in RANSAC such that its computation time is close to the GNC-based methods. For the other baselines, we use their default settings. Results on ModelNet [59], 3DMatch [65] and KITTI [29] are shown in Tab. **4**. Each experiment was repeated 10 times. Across all datasets, we observe that 1) SAC-GNC++ is the most accurate at a time cost, 2) SAC-GNC is second, being close to



(a) Data (≈ 90% outliers)

(b) **SAC-GNC** (Ours)

Figure 3. 3D registration example in an extreme outlier rate scenario from 3DMatch [65] (data is from the ↓ inliers set).



(a) INTEL (2D, real-world data)

(b) CSAIL (2D, real-world data)

Figure 4. Pose graph optimization results on (a) INTEL and (b) CSAIL datasets. For each sequence, the left image displays the "Initial trajectory", the trajectory estimated by SAC-GNC with 50% outlier loops, and the "Reference" trajectory. The right image shows the mAA at 25 cm (dashed line) and 1 m (solid line) for the trajectory error (higher mAA is better) over the average computational time, for a percentage of randomly perturbed loops varying between 10% (leftmost dot) to 90% (rightmost dot).

SAC-GNC++ in accuracy but faster, and 3) SAC-GNC has the best trade-off between accuracy and efficiency. While RANSAC is fast and performs well for high inlier rates, it has a large performance drop when dealing with considerable outliers. Among the GNC-based approaches, the adaptive strategies (ours and GNCp) always outperform the fixed annealing strategies (FGR and TEASER++). Figure **3** shows an example of 3DMatch data (↓ inliers) and the respective result obtained by SAC-GNC.

5.2. Pose graph optimization problem

Next, we consider the PGO problem. PGO optimizes N global pose transformations $v_i \in \mathrm{SE}(2), i = 1, \cdots N$ from relative measurements $\tilde{e}_{i,j} \in \mathrm{SE}(2)$ acquired along some trajectory. Following [27], we define the problem as the one in Eq. **2**, with residuals given by

$$r(\tilde{e}_{i,j}, v_i, v_j) = \|\log\left(\tilde{e}_{i,j}^{-1}\, v_i^{-1}\, v_j\right)^{\vee}\|_{\Sigma}, \tag{9}$$

where $\log(\cdot)^{\vee}$ brings an element of SE(2) to its tangent space and $\Sigma \in \mathbb{R}^{3\times 3}$ is a covariance matrix. In this formulation, the data and model are given by $\boldsymbol{x} = \{\tilde{e}_{i,j}\}$ and $\theta = \{v_i\}$, respectively. Note that this is a large-scale problem, where the model θ to be updated consists of all camera poses. We solve Eq. **2** with residuals in Eq. **9** using GTSAM's Levenberg–Marquardt (LM) optimizer in [24].

We ran our approach on several real-world SLAM benchmark datasets. Due to space limitations, we show the INTEL [15] and CSAIL [16] sequences and provide more (2D/3D) in the supplementary material. For generating outliers, we follow [60]. We keep the odometry measurements, named "Initial trajectory", and perturb loops with random transformations. The rate of loops perturbed varies from 10% to 90%, repeating each 100 times (each run uses a new perturbed graph). Since there is no ground truth, the reference is the result obtained for 0% outliers, which we denote as "Reference". As in [27], the inlier threshold is 0.5.

The current state-of-the-art approach for PGO is [60]. We use its GTSAM [24] implementation with the Geman-McClure loss and default settings, denoted as GTSAM-GNC. SAC-GNC and GTSAM-GNC only differ in the σ initialization, σ update, and stopping criteria. Results comparing our approach and GTSAM-GNC are provided in Fig. **4**. Our main observation relates to efficiency, as SAC-GNC and SAC-GNC++ are much faster than GTSAM-GNC. Concerning accuracy, all methods have similar results until around 50% outliers. Beyond 50% outliers, our approach outperforms GTSAM-GNC.

6. Conclusion

We propose SAC-GNC, a novel algorithm for adaptive annealing in GNC. It employs an annealing sample and consensus strategy by testing various annealing factors at each GNC iteration and scoring each computed model to decide the most promising shape parameter to explore. Additionally, we propose new stopping criteria and shape initialization. Extensive results in two computer vision problems demonstrate that our solution is the most robust and efficient than the baselines. Lastly, this paper demonstrates that combining sample and consensus into GNC offers advantages over previous GNC-only approaches.

Acknowledgments

Valter Piedade was supported by Mitsubishi Electric Research Laboratories, LARSyS funding (DOI: 10.54499/LA/P/0083/2020), and Sustainable Stone by Portugal (co-financed by the Recovery and Resilience Plan of the European Union). Pedro Miraldo was supported exclusively by Mitsubishi Electric Research Laboratories.

References

[1] Daniel Barath and Jiri Matas. Graph-cut ransac. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 6733–6741, 2018. 3

[2] Daniel Barath, Maksym Ivashechkin, and Jiri Matas. Progressive napsac: sampling from gradually growing neighborhoods. *arXiv preprint arXiv:1906.02295*, 2019.

[3] Daniel Barath, Jiri Matas, and Jana Noskova. Magsac: marginalizing sample consensus. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 10197–10205, 2019. 5

[4] Daniel Barath, Jana Noskova, Maksym Ivashechkin, and Jiri Matas. Magsac++, a fast, reliable and accurate robust estimator. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 1304–1312, 2020. 3

[5] Daniel Barath, Dmytro Mishkin, Luca Cavalli, Paul-Edouard Sarlin, Petr Hruby, and Marc Pollefeys. Affineglue: Joint matching and robust estimation. *arXiv preprint arXiv:2307.15381*, 2023. 3

[6] Daniel Barath, Dmytro Mishkin, Michal Polic, Wolfgang Förstner, and Jiri Matas. A large-scale homography benchmark. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 21360–21370, 2023. 6

[7] Michael J Black and Anand Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *Int. J. Comput. Vis. (IJVC)*, 19(1):57–91, 1996. 1, 3

[8] Andrew Blake and Andrew Zisserman. *Visual reconstruction*. MIT press, 1987. 1, 3

[9] Grigoriy Blekherman, Pablo A Parrilo, and Rekha R Thomas. *Semidefinite optimization and convex algebraic geometry*. SIAM, 2012. 2

[10] Eric Brachmann and Carsten Rother. Learning less is more-6d camera localization via 3d surface regression. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 4654–4662, 2018. 3

[11] Eric Brachmann and Carsten Rother. Neural-guided ransac: Learning where to sample model hypotheses. In *IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pages 4322–4331, 2019. 3

[12] Jesus Briales and Javier Gonzalez-Jimenez. Convex global 3d registration with lagrangian duality. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 4960–4969, 2017. 2

[13] Jesus Briales, Laurent Kneip, and Javier Gonzalez-Jimenez. A certifiably globally optimal solution to the non-minimal relative pose problem. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 145–154, 2018. 2

[14] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Trans. Robotics (T-RP)*, 32(6): 1309–1332, 2016. 2

[15] Luca Carlone and Andrea Censi. From angular manifolds to the integer lattice: Guaranteed orientation estimation with application to pose graph optimization. *IEEE Trans. Robotics (T-RP)*, 30(2):475–492, 2014. 8

[16] Luca Carlone, Rosario Aragues, José A Castellanos, and Basilio Bona. A fast and accurate approximation for planar pose graph optimization. *The Int. J. of Robotics Research*, 33(7):965–987, 2014. 8

[17] Luca Carlone, David M Rosen, Giuseppe Calafiore, John J Leonard, and Frank Dellaert. Lagrangian duality in 3d slam: Verification techniques and optimal solutions. In *IEEE Int. Conf. Intell. Robots Systems (IROS)*, pages 125–132, 2015. 2

[18] Luca Carlone, Giuseppe C Calafiore, Carlo Tommolillo, and Frank Dellaert. Planar pose graph optimization: Duality, optimal solutions, and verification. *IEEE Trans. Robotics (T-RP)*, 32(3):545–565, 2016. 2

[19] Avishek Chatterjee and Venu Madhav Govindu. Robust relative rotation averaging. *IEEE Trans. Pattern Anal. Mach. Intell. (T-PMAI)*, 40(4):958–972, 2018. 1, 3

[20] Hongkai Chen, Zixin Luo, Jiahui Zhang, Lei Zhou, Xuyang Bai, Zeyu Hu, Chiew-Lan Tai, and Long Quan. Learning to match features with seeded graph matching network. In *IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pages 6301–6310, 2021. 3

[21] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2019. 3, 6

[22] Ondrej Chum and Jiri Matas. Matching with prosac-progressive sample consensus. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 220–226, 2005. 3

[23] Ondrej Chum, Jiri Matas, and Josef Kittler. Locally optimized ransac. In *Joint Pattern Recog. Symposium*, pages 236–243, 2003. 3

[24] Frank Dellaert and GTSAM Contributors. borglab/gtsam. `https://github.com/borglab/gtsam`, 2022. 8

[25] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest

point detection and description. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 224–236, 2018. 3

[26] Yaqing Ding, Jian Yang, Viktor Larsson, Carl Olsson, and Kalle Åström. Revisiting the p3p problem. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 4872–4880, 2023. 3

[27] Kevin J Doherty, Ziqi Lu, Kurran Singh, and John J Leonard. Discrete-continuous smoothing and mapping. *IEEE Robotics Automation Letters (R-AL)*, 7(4): 12395–12402, 2022. 8

[28] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 3, 5, 7

[29] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2012. 6, 7

[30] Richard I Hartley and Fredrik Kahl. Global optimization through rotation space search. *Int. J. Comput. Vis. (IJVC)*, 82(1):64–79, 2009. 2

[31] Berthold K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Amer*, 4(4):629–642, 1987. 3

[32] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. Predator: Registration of 3d point clouds with low overlap. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 4267–4276, 2021. 3, 6

[33] Viktor Larsson, Zuzana Kukelova, and Yinqiang Zheng. Making minimal solvers for absolute pose estimation compact and robust. In *IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pages 2316–2324, 2017. 3

[34] Philipp Lindenberger, Paul-Edouard Sarlin, Viktor Larsson, and Marc Pollefeys. Pixel-perfect structure-from-motion with featuremetric refinement. In *IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pages 5987–5997, 2021. 3

[35] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed. *arXiv preprint arXiv:2306.13643*, 2023. 3

[36] Kirk MacTavish and Timothy D Barfoot. At all costs: A comparison of robust cost functions for camera correspondence outliers. In *IEEE Conf. Computer Robot Vision*, pages 62–69, 2015. 3

[37] Joshua G Mangelson, Jinsun Liu, Ryan M Eustice, and Ram Vasudevan. Guaranteed globally optimal planar pose graph and landmark slam via sparse-bounded sums-of-squares programming. In *IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 9306–9312, 2019. 2

[38] Desire L Massart, Leonard Kaufman, Peter J Rousseeuw, and Annick Leroy. Least median of squares: a robust method for outlier and model error detection in regression and calibration. *Analytica Chimica Acta*, 187:171–179, 1986. 5

[39] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Anal. Mach. Intell. (T-PMAI)*, 26(6):756–770, 2004. 3

[40] Carl Olsson, Fredrik Kahl, and Magnus Oskarsson. Branch-and-bound methods for euclidean registration problems. *IEEE Trans. Pattern Anal. Mach. Intell. (T-PMAI)*, 31(5):783–794, 2008. 2

[41] Linfei Pan, Dániel Baráth, Marc Pollefeys, and Johannes L Schönberger. Global structure-from-motion revisited. In *Eur. Conf. Comput. Vis. (ECCV)*, 2024. 2

[42] Valter Piedade and Pedro Miraldo. Bansac: A dynamic bayesian network for adaptive sample consensus. In *IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pages 3738–3747, 2023. 3

[43] Anand Rangarajan and Rama Chellappa. Generalized graduated nonconvexity algorithm for maximum a posteriori image estimation. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 127–133, 1990. 1, 3

[44] Jerome Revaud, Philippe Weinzaepfel, César De Souza, Noe Pion, Gabriela Csurka, Yohann Cabon, and Martin Humenberger. R2d2: repeatable and reliable detector and descriptor. *arXiv preprint arXiv:1906.06195*, 2019. 3

[45] David M Rosen, Luca Carlone, Afonso S Bandeira, and John J Leonard. Se-sync: A certifiably correct algorithm for synchronization over the special euclidean group. *Int. Journal Robotics Research*, 38(2-3):95–125, 2019. 2

[46] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson, fourth edition, 2020. 4, 5

[47] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 3212–3217. IEEE, 2009. 6

[48] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 4938–4947, 2020. 3

[49] Paul-Edouard Sarlin, Ajaykumar Unagar, Mans Larsson, Hugo Germain, Carl Toft, Viktor Larsson, Marc Pollefeys, Vincent Lepetit, Lars Hammarstrand, Fredrik Kahl, et al. Back to the feature: Learning robust camera localization from pixels to pose. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 3247–3257, 2021. 3

[50] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 4104–4113, 2016. 2

[51] Yan Shi, Jun-Xiong Cai, Yoli Shavit, Tai-Jiang Mu, Wensen Feng, and Kai Zhang. Clustergnn: Cluster-based coarse-to-fine graph neural network for efficient feature matching. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 12517–12526, 2022. 3

[52] Chitturi Sidhartha, Lalit Manam, and Venu Madhav Govindu. Adaptive annealing for robust geometric estimation. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 21929–21939, 2023. 1, 2, 3, 6, 7

[53] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 8922–8931, 2021. 3

[54] Philip HS Torr and Andrew Zisserman. Mlesac: A new robust estimator with application to estimating image geometry. *Comput. Vis. Image Understanding (CVIU)*, 78(1):138–156, 2000. 5, 6

[55] Vasileios Tzoumas, Pasquale Antonante, and Luca Carlone. Outlier-robust spatial perception: Hardness, general-purpose algorithms, and guarantees. In *IEEE Int. Conf. Intell. Robots Systems (IROS)*, pages 5383–5390, 2019. 2

[56] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. Pattern Anal. Mach. Intell. (T-PMAI)*, 13 (04):376–380, 1991. 2, 6

[57] Yue Wang and Justin M Solomon. Prnet: Self-supervised learning for partial-to-partial registration. *MeurIPS*, 32, 2019. 3

[58] Tong Wei, Jiri Matas, and Daniel Barath. Adaptive reordering sampler with neurally guided magsac. In *IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pages 18163–18173, 2023. 3

[59] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 1912–1920, 2015. 6, 7

[60] Heng Yang, Pasquale Antonante, Vasileios Tzoumas, and Luca Carlone. Graduated non-convexity for robust spatial perception: From non-minimal solvers to global outlier rejection. *IEEE Robotics Automation Letters (R-AL)*, 5(2):1127–1134, 2020. 1, 2, 3, 6, 8

[61] Heng Yang, Jingnan Shi, and Luca Carlone. Teaser: Fast and certifiable point cloud registration. *IEEE Trans. Robotics (T-RP)*, 37(2):314–333, 2020. 2, 3, 7

[62] Ming Ye, Robert M Haralick, and Linda G Shapiro. Estimating piecewise-smooth optical flow with global matching and graduated optimization. *IEEE Trans. Pattern Anal. Mach. Intell. (T-PMAI)*, 25(12):1625–1630, 2003. 3

[63] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to find good correspondences. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 2666–2674, 2018. 3, 6

[64] Christopher Zach and Guillaume Bourmaud. Descending, lifting or smoothing: Secrets of robust cost optimization. In *Eur. Conf. Comput. Vis. (ECCV)*, pages 547–562, 2018. 3

[65] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017. 6, 7, 8

[66] Jiahui Zhang, Dawei Sun, Zixin Luo, Anbang Yao, Lei Zhou, Tianwei Shen, Yurong Chen, Long Quan, and Hongen Liao. Learning two-view correspondences and geometry using order-aware network. In *IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pages 5845–5854, 2019. 3

[67] Ji Zhao, Wanting Xu, and Laurent Kneip. A certifiably globally optimal solution to generalized essential matrix estimation. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 12034–12043, 2020. 2

[68] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In *Eur. Conf. Comput. Vis. (ECCV)*, pages 766–782. Springer, 2016. 2, 3, 7

[69] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018. 7