2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Robust Two-View Geometry Estimation with Implicit Differentiation

Vladislav Pyatov[1], Iaroslav Koshelev[2] and Stamatios Lefkimmiatis[3].

Abstract— **We present a novel two-view geometry estimation framework which is based on a differentiable robust loss function fitting. We propose to treat the robust fundamental matrix estimation as an implicit layer, which allows us to avoid backpropagation through time and significantly improves the numerical stability. To take full advantage of the information from the feature matching stage we incorporate learnable weights that depend on the matching confidences. In this way our solution brings together feature extraction, matching and two-view geometry estimation in a unified end-to-end trainable pipeline. We evaluate our approach on the camera pose estimation task in both outdoor and indoor scenarios. The experiments on several datasets show that the proposed method outperforms both classic and learning-based state-of-the-art methods by a large margin. The project webpage is available at: `https://github.com/VladPyatov/ihls`**

I. INTRODUCTION

Nowadays, 3D computer vision has received widespread attention and has found application in a wide range of tasks, including scene reconstruction [1], autonomous driving [2] and neural rendering [3]. To deal with such tasks a first necessary step is to recover the position and orientation of each camera in the scene or equivalently to solve the Camera Pose Estimation (CPE) problem [4], [5].

Given a set of input images, the solution of CPE typically consists of three essential steps. First, putative keypoint correspondences are established through the feature extraction [6]–[9] and matching [10], [11] stages. The second step involves the recovery of the two-view geometry [12], [13] by estimating the relative camera poses between image pairs. This estimate provides a good initial solution that is later refined by bundle adjustment [14] in the final stage. The feature matching and bundle adjustment problems have received a lot of attention over the past few years, especially with the advent of learning-based approaches. At the same time two-view geometry estimation has not been widely studied.

The two-view geometry of an image pair is usually represented either with an essential [15] or (in the uncalibrated camera setup) a fundamental matrix [16] that define an algebraic representation of the geometric epipolar constraint [12]. These matrices can be estimated from a set of putative keypoint correspondences using the *least squares* approach. However, due to occlusions, repetitive patterns, illumination changes and poor textures present in a scene, the feature matching stage may produce outliers, mismatched pairs that affect the solution and, above all, lead to an incorrect geometry representation. Given that the least squares method works under the assumption that the noise in the matches obeys a Gaussian distribution, it is unable to efficiently handle severe outliers and can lead to poor estimation results.

One way to deal with outliers is to utilize RANSAC-based estimators [17]–[22]. The main strategy followed by these methods is based on the hypothesize-and-verify approach. Specifically, after the feature matching stage, minimal subsets of the putative matches are randomly sampled and the corresponding two-view geometries are estimated with either the 8- [15] 7- [23] or 5-point [24] algorithms in order to form a pool of model hypotheses. Then the generated hypotheses are scored by counting the number of inliers, *i.e.*, matches with residual error below a selected threshold. Finally, the best model hypothesis is further refined by using all the inlier correspondences. Despite their effectiveness, the non-deterministic nature of RANSAC-based estimators precludes efficient end-to-end training when neural networks are involved in the overall pipeline. As a consequence, taking full advantage of the information available from the feature extraction and matching stages becomes challenging.

Another line of work that aims to circumvent the problems appearing due to outliers, is devoted to the introduction of robust loss functions [25], [26], a specific type of M-estimators [27]. According to this approach, the solution to the two-view geometry problem is obtained by optimizing an objective function that is less sensitive to the presence of outliers. In general, the task is formulated as a non-convex and possibly non-smooth optimization problem that is typically solved employing iterative numerical algorithms [28]. In general, given the non-convex nature of the objective, the solution is not guaranteed to attain the global minimum, but in practice if the fraction of outliers is relatively small, then the recovered two-view geometry is more precise compared to the one estimated with RANSAC-based methods. On the other hand, when the outlier pairs start dominating the correspondences, the solution becomes sub-optimal and hurts the performance of the downstream tasks.

The learning-based approach in [29] tries to alleviate this problem by aiming to implicitly learn a robust loss function. However, this method faces the important limitation of having to back-propagate through unrolled layers that involve the singular value decomposition (SVD). This leads both to an increased memory footprint during training as well as numerical instabilities mostly from the back-propagation through SVD. Therefore, the authors of [29] limit the number of the employed unrolled layers to just a few, which in turn

[1]Vladislav Pyatov is with the Skolkovo Institute of Science and Technology (Skoltech), Center for AI Technology, Russia, vladislav.pyatov@skoltech.ru
[2] Iaroslav Koshelev is with the AI Foundation and Algorithm Lab, Russia.
[3]Stamatios Lefkimmiatis is with the MTS AI Group, Russia. s.lefkimmiatis@mts.ai

hinders the final performance of their approach.

In this work we revisit the direction of robust loss function fitting and propose a novel approach for outlier-robust two-view geometry estimation problem. First, we adopt a specific parametric form for the robust loss that exhibits several important benefits both in terms of robustness to outliers and efficiency in training. To successfully minimize this non-convex loss we introduce the Iterative Homogeneous Least Squares (IHLS) solver that takes as input putative keypoint correspondences and outputs an estimate of the underlying two-view geometry. To avoid back-propagation through IHLS with unrolling, we take advantage of the convergence guaranties of our solver and utilize the implicit function theorem that allow us to perform back-propagation without the need to save any intermediate results from the forward pass or back-propagate through the SVD layer, which is an essential part of our optimization algorithm. Further, to downplay the role of the outliers, we develop a learnable framework on top of the IHLS solver and utilize a weight prediction network that re-estimates matching confidences in a recurrent fashion based on the matches, residuals of the current solution and auxiliary information from the feature matching stage. We then chain our framework with the LoFTR [11] dense feature matching transformer to obtain a fully differentiable robust camera pose estimation pipeline. To summarize, our main contributions are as follows:

- We employ a robust parametric loss function to deal with the fundamental matrix estimation problem and propose the Iterative Homogeneous Least Squares solver which can efficiently recover a solution of the underlying minimization problem.
- The convergence guaranties of IHLS further allows us to introduce an efficient end-to-end training strategy using implicit backpropagation, which significantly reduces the memory requirements and improves the stability of training compared to alternative approaches that employ unrolled networks.
- We propose an end-to-end trainable camera pose estimation pipeline that brings together feature extraction, matching and robust two-view geometry estimation.

II. RELATED WORK

A. Feature extraction and matching

Hand-crafted local features, such as SIFT [6] and ORB [30], combined with a mutual nearest neighbour search or Lowe's ratio test [6] have been considered as the dominant correspondence estimation approach for a long time. Nevertheless, with the advent of deep learning the components of the classic pipeline have been largely revisited. In particular, learning-based local features [7], [8] show impressive results in challenging conditions of viewpoint change and poor texture. SuperGlue [10] proposes a learning-based approach for local feature matching on top of the sparse SuperPoint [8] features, while the recently introduced LightGlue [31] further reduces it's computational complexity. LoFTR [11] and its successors [32] take advantage of dense feature extraction and propose a detector-free design.

B. RANSAC-based estimators

RANSAC-based approaches either try to improve hypothesis scoring of the original RANSAC [17] or propose end-to-end trainable counterparts. MLESAC [18] chooses the solution that maximizes the likelihood rather than just the number of inlier pairs and thus is less sensitive to the inlier threshold. LO-RANSAC [19] applies local optimization to the so-far-the-best model hypothesis. DEGENSAC [20] detects degenerate configurations and rejects models that are incompatible with the scene geometry. MAGSAC [21] proposes to marginalize the quality function over a range of noise levels to eliminate the threshold from the model quality calculation. State-of-the-art approach MAGSAC++ [22] proposes a novel marginalization procedure and introduces a new model quality function that does not require the inlier-outlier decision. On the other hand, inspired by the success of reinforcement learning, DSAC [33] methods introduce a first learning-based alternative and incorporate REINFORCE [34] algorithm to make the hypothesis selection procedure differentiable. Neural-guided RANSAC [35] makes a step forward and proposes learned guidance of hypothesis sampling. ∇-RANSAC [36] revises the learning-based problem formulation and incorporates the Gumbel Softmax [37] relaxation to estimate the gradients in the sampling distribution.

C. Robust loss functions

Robust loss functions serve as an alternative strategy for outlier-robust model fitting and is the main focus of this work. Since their introduction in the context of M-estimators [27], their applicability to the two-view geometry estimation [38] has been proven. The aforementioned MAGSAC++ [22] reformulates the model fitting problem as an M-estimation and solves it with an iteratively reweighted least squares (IRLS) procedure. To downplay the role of outliers Ranfl and Koltun [29] proposed to cast the robust function learnable and simulated the IRLS procedure in their Deep Fundamental Matrix Estimation (DFE) approach. In attempts to overcome the limitations of IRLS, Ikami *et al*. [39] developed a novel objective function and proposed the iteratively reweighted eigenvalues minimization (IREM) for its optimization.

III. METHOD

The two-view geometry of an image pair can be represented either with the essential matrix $\mathbf{E}$ if the camera intrinsics are known or with the fundamental matrix $\mathbf{F}$ in the more general un-calibrated setup. The minimal solver for the essential matrix is the 5-point algorithm [24], while for fundamental matrix estimation there are 8- and 7-point algorithms [15], [23]. In our work we use the 8-point problem formulation as it is applicable to both the fundamental and essential matrix estimation tasks. Without limiting the generality of our approach, further derivations are provided for the fundamental matrix.

Fig. 1 depicts the overview of our proposed approach. Given the source image $\boldsymbol{I}_S$ and target image $\boldsymbol{I}_T$, it estimates the fundamental matrix $\mathbf{F}$ representing the two-view

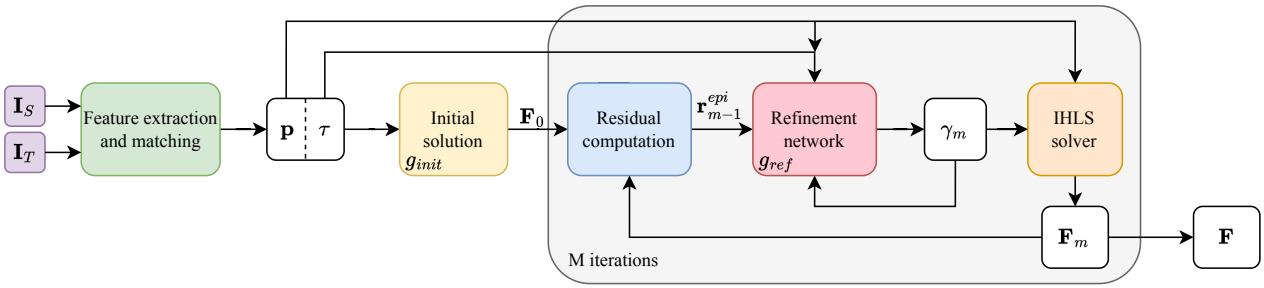

Fig. 1. Overview of the proposed framework. The pair of source and target images $\boldsymbol{I}_S, \boldsymbol{I}_T$ is passed through the LoFTR feature extraction and matching module to predict the set of correspondences $\boldsymbol{p}$ and a context vectors $\boldsymbol{\tau}$. Both are then used to initialize and guide the fundamental matrix estimation via the proposed recurrent strategy consisting of refinement network and IHLS solver. For the detailed description of each component please refer to Sec. III.

geometry of the image pair. Our framework consists of the following modules:

- **Feature extraction and matching.** Given the image pair, this module estimates the set of keypoint correspondences $\boldsymbol{p}$ and the corresponding additional information $\boldsymbol{\tau}$ describing the quality of matches.
- **Initial solution.** The module computes the initial solution $\mathbf{F}_0$ with a weighted 8-point algorithm. The weights $\boldsymbol{\gamma}_0$ are estimated with a lightweight neural network g_{init} that takes as input $\boldsymbol{p}$ and $\boldsymbol{\tau}$.
- **Outlier-robust two-view geometry refinement.** In this recurrent block, the epipolar residual $\boldsymbol{r}^{epi}_{m-1}$ is first computed, based on the solution $\mathbf{F}_{m-1}$ from the previous iteration. Next, the residual $\boldsymbol{r}^{epi}_{m-1}$ along with correspondences $\boldsymbol{p}$ and additional information $\boldsymbol{\tau}$ serves as input to the refinement network g_{ref} that updates confidence weights $\boldsymbol{\gamma}_m$ of the matches. The weights are then utilized in our outlier-robust IHLS solver to estimate the refined solution $\mathbf{F}_m$.

In the following sections we provide the detailed description of the proposed pipeline.

A. Feature extraction and matching

Given the pair of images $(\boldsymbol{I}_S, \boldsymbol{I}_T)$, the first step in recovering the two-view geometry is to extract distinctive features from both images and subsequently match them. To demonstrate the potential of end-to-end training with our two-view geometry estimation modules, as a backbone model for this task we utilize a dense feature matching transformer LoFTR [11]. It gets as input the image pair, performs feature extraction, coarse-level matching and then refines confident matches to a sub-pixel level. As a result, it outputs a set of matches $\boldsymbol{p}$ as well as their confidence levels $\boldsymbol{w} \in [0, 1]$.

For the set of matched keypoints $\boldsymbol{p}$, we compute the following quantities as the additional information $\boldsymbol{\tau} \in \mathbb{R}^{N\times 4}$:

1) coarse-level matching confidences $\boldsymbol{w}$
2) L_2 coarse-level descriptor distances $\boldsymbol{d}_2$
3) $cosine$ coarse-level descriptor distances $\boldsymbol{d}_{cos}$
4) fine-level standard deviation of match position in the second image $\boldsymbol{\sigma}$

It is worth noting that our robust fitting method does not depend on the specific type of feature extraction or matching, and can be successfully combined with both classic and learnable pipelines.

B. Initial solution

If a 3D point is observed in a pair of images $\boldsymbol{I}_S, \boldsymbol{I}_T$, where it has the homogeneous coordinates $\mathbf{x} = (x, y, 1)$ and $\mathbf{x}' = (x', y', 1)$, then such coordinates satisfy the so called epipolar constraint [12]:

$$\mathbf{x}'^T \mathbf{F} \mathbf{x} = (\mathbf{x}'^T \otimes \mathbf{x}^T) \operatorname{vec}(\mathbf{F}) = 0. \tag{1}$$

In this equation $\mathbf{F} \in \mathbb{F} \equiv \{\mathbf{F} \in \mathbb{R}^{3\times 3} \mid rank(\mathbf{F}) = 2\}$ represents the fundamental matrix, $\otimes$ is the Kronecker product and $\operatorname{vec}(\cdot)$ is the vectorization operator.

Given a set of N keypoint correspondences $\boldsymbol{p} = \{(\mathbf{x}_n, \mathbf{x}'_n)\}_{n=1}^N$, this constraint can be expressed compactly in the form of a homogeneous linear system

$$\boldsymbol{r} \equiv \mathbf{A}\boldsymbol{f} = 0, \tag{2}$$

where $\boldsymbol{f} = \operatorname{vec}(\boldsymbol{F})$, $\mathbf{A} \in \mathbb{R}^{N\times 9}$ and each row

$$\mathbf{A}_n = [x'_n x_n, x'_n y_n, x'_n, y'_n x_n, y'_n y_n, y'_n, x_n, y_n, 1] \tag{3}$$

corresponds to the n-th matching pair $(\mathbf{x}_n, \mathbf{x}'_n)$.

Then the 8-point algorithm estimates the fundamental matrix as a solution of the following *Least Squares* problem:

$$\boldsymbol{f}^\star = \underset{\|\boldsymbol{f}\|_2=1}{\arg\min} \|\mathbf{A}\boldsymbol{f}\|_2^2 = \underset{\|\boldsymbol{f}\|_2=1}{\arg\min} \sum_{n=1}^{N} r_n^2 = \boldsymbol{u}_{min}(\mathbf{A}), \tag{4}$$

where $r_n \equiv r(\mathbf{x}_n, \mathbf{x}'_n, \boldsymbol{f}) \coloneqq (\mathbf{A}\boldsymbol{f})_n$ is the residual error for the n-th keypoints pair and $\boldsymbol{u}_{min}(\mathbf{A})$ is the singular vector of $\mathbf{A}$ that corresponds to its smallest singular value.

The solution of (4) does not necessarily lead to a fundamental matrix of rank 2, as required. Therefore, the final estimate is obtained by projecting $\boldsymbol{f}$ on the space of rank-2 matrices. Specifically if we express the estimated matrix using SVD as $\boldsymbol{f} = \mathbf{U} \operatorname{diag}(\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2, \boldsymbol{\sigma}_3) \mathbf{V}^\mathsf{T}$, with the singular values being in decreasing order, then its projection is equal to $\mathbf{F} = \Pi(\boldsymbol{f}) \coloneqq \mathbf{U} \operatorname{diag}(\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2, \mathbf{0}) \mathbf{V}^\mathsf{T}$.

We are interested in outlier-robust solution, however as noticed in [39] robust estimation algorithms are sensitive to the initialization. To alleviate this problem, we first estimate the initial solution $\mathbf{F}_0$ and then perform the iterative refinement as described in the next section.

To estimate the initial solution we solve (4) with weights $\boldsymbol{\gamma}_0 \in \mathbb{R}^N$ to suppress the influence of outliers:

$$\boldsymbol{f}_0 = \underset{\|\boldsymbol{f}\|_2=1}{\arg\min} \|\operatorname{diag}(\boldsymbol{\gamma}_0) \mathbf{A}\boldsymbol{f}\|_2^2 \tag{5}$$

The weights $\boldsymbol{\gamma}_0$ are estimated with the PointNet-like [40] neural network $g_{init}(\boldsymbol{p}, \boldsymbol{\tau})$ that processes $\boldsymbol{p}$ and $\boldsymbol{\tau}$ with interleaving 1D Convolution and Context Normalization layers. The network architecture is close to [29] with the exception that the last layer is the sigmoid function, instead of softmax.

C. Outlier-Robust Two-View Geometry Refinement

Minimizing a quadratic function of the residual error can over-penalize the outliers and lead to a sub-optimal solution. For this reason, alternative robust loss functions have been studied. In this setup the solution can be recovered as:

$$\boldsymbol{f}^{\star} = \underset{\|\boldsymbol{f}\|_2=1}{\arg\min}\, \rho\left(\mathbf{A}\boldsymbol{f}; \boldsymbol{\theta}\right) = \underset{\|\boldsymbol{f}\|_2=1}{\arg\min} \sum_{n=1}^{N} \rho\left(r_n; \boldsymbol{\theta}\right), \quad (6)$$

where ρ is a robust loss function of the residual error $\boldsymbol{r}$ with parameters $\boldsymbol{\theta}$.

The use of a robust function ρ should lead to solutions that exhibit zero residual errors for all the inliers and non-zero residuals otherwise. Moreover, for outliers we expect the robust loss to be agnostic to the level of residual error, which is not the case if the quadratic loss is employed in (4). In other words, we aim for a solution $\boldsymbol{f}^{\star}$ which leads to a vector of residual errors $\boldsymbol{r}$ that attain a certain level of sparsity, related to the total number of inliers.

To accomplish this, we could resort to the ℓ_1 penalty, which corresponds to the tightest convex relaxation of the sparsity measure [41], or to non-convex penalties that can lead to an even better approximation. Among those is the ℓ_p^p pseudo-norm for $p \in (0, 1)$ [42]:

$$\rho\left(\boldsymbol{r}; \epsilon, p\right) = \sum_{n=1}^{N} \left(r_n^2 + \epsilon\right)^{p/2}, \quad (7)$$

where $\epsilon > 0$ guarantees numerical stability.

Nevertheless, in all these cases certain outliers can still be over-penalized, and thus affect the accuracy of the estimation. To downplay the role of such an outliers, we introduce a weighted extension of ℓ_p^p, which we define as

$$\rho\left(\tilde{\boldsymbol{r}}; \epsilon, p\right) = \rho\left(\boldsymbol{\gamma} \odot \boldsymbol{r}; \epsilon, p\right) = \sum_{n=1}^{N} \left(\left(\gamma_n r_n\right)^2 + \epsilon\right)^{p/2}. \quad (8)$$

Here $\boldsymbol{\gamma}$ is a vector of non-negative weights and by $\odot$ we denote the point-wise multiplication.

The solution of the optimization problem (6) with the specified robust loss function (8) is then estimated in our IHLS solver, the detailed explanation of which is provided in the next section:

$$\boldsymbol{f}^{\star} = \text{IHLS}(\boldsymbol{p}, \boldsymbol{\gamma}; \epsilon, p) = \text{IHLS}(\mathbf{A}; \epsilon, p) \quad (9)$$

For simplicity, hereinafter we assume that for the observation matrix it holds $\mathbf{A} = \mathbf{A}(\boldsymbol{p}, \boldsymbol{\gamma})$. Indeed, it can be shown that

$$\tilde{\boldsymbol{r}} \equiv \boldsymbol{\gamma} \odot \boldsymbol{r} = \boldsymbol{\gamma} \odot \left(\mathbf{A}\boldsymbol{f}\right) = \text{diag}\left(\boldsymbol{\gamma}\right)\mathbf{A}\boldsymbol{f} = \tilde{\mathbf{A}}\boldsymbol{f} \quad (10)$$

The optimal weight $\boldsymbol{\gamma}$ could be directly derived from the optimal residual error $\tilde{\boldsymbol{r}}$. On the other hand, the optimal residual can not be attained without clear separation of the correspondences into inliers and outliers. Therefore, we propose to adjust the weights $\boldsymbol{\gamma}$ in a recurrent manner.

Specifically, given the solution $\mathbf{F}_{m-1}$ from the previous iteration $m-1$, we first calculate the epipolar residual $\boldsymbol{r}_{m-1}^{epi}$ as the *Symmetric Epipolar Distance* [12]. The $\boldsymbol{r}^{epi}$ can be seen as the normalized version of $\boldsymbol{r}$ to preserve the constrained input range for the network. We then process $\boldsymbol{r}_{m-1}^{epi}$ with the network $g_{ref}(\boldsymbol{r}_{m-1}^{epi}, \boldsymbol{p}, \boldsymbol{\tau})$ to obtain the updated weights $\boldsymbol{\gamma}_m$. The architecture of the g_{ref} is similar to g_{init}. Note also that we employ a shared weights setup, so in each iteration we use the same network g_{ref}.

With the new weights $\boldsymbol{\gamma}_m$ we find the outlier-robust solution $\mathbf{F}_m$ with our IHLS solver and proceed to the next iteration. The final solution $\mathbf{F}$ and the weights $\boldsymbol{\gamma}$ are then simply the output of the last iteration. Based on our experiments, we observed that an initialization step and 2 iterations of the recurrent unit are enough for the convergence of our framework.

D. Iterative Homogeneous Least Squares Solver

In this section we describe in detail our approach for solving the optimization problem (6) with the proposed outlier-robust loss function defined in Eq. (8). To proceed, we utilize the following inequality [43]:

$$|f|^p \leq \frac{p}{2}\frac{f^2}{\beta^{2-p}} + \frac{2-p}{2}\beta^p, \quad (11)$$

which holds true $\forall f \in \mathbb{R}$, $\beta > 0$ and $0 < p \leq 2$.

Let us now consider the positive scalar $\sqrt{f_n^2 + \epsilon}$ with $\epsilon > 0$ being a small positive constant and plug it in (11). Then, we obtain

$$\left(f_n^2 + \epsilon\right)^{\frac{p}{2}} \leq \frac{p}{2}\frac{f_n^2}{\beta_n^{2-p}} + \frac{p\epsilon}{2}\beta_n^{p-2} + \frac{2-p}{2}\beta_n^p. \quad (12)$$

The above inequality is closed under summation and, thus, it further holds that:

$$\begin{aligned}\rho\left(\boldsymbol{f}; \epsilon, p\right) = \sum_{n=1}^{N}\left(f_n^2 + \epsilon\right)^{\frac{p}{2}} &\leq \frac{p}{2}\boldsymbol{f}^{\mathsf{T}}\text{diag}^{p-2}\left(\boldsymbol{\beta}\right)\boldsymbol{f} \\ &+ \sum_{n=1}^{N}\left(\frac{p\epsilon}{2}\beta_n^{p-2} + \frac{2-p}{2}\beta_n^p\right) = \psi\left(\boldsymbol{\beta}\right). \quad (13)\end{aligned}$$

Moreover, by noting that all the terms of $\psi\left(\boldsymbol{\beta}\right)$ are strictly positive, we end up with the following identity

$$\rho\left(\boldsymbol{f}; \epsilon, p\right) = \min_{\boldsymbol{\beta} \in \mathbb{R}_+^N} \psi\left(\boldsymbol{\beta}\right), \quad (14)$$

which can be easily verified by differentiating w.r.t $\boldsymbol{\beta}$. We have, thus, managed to obtain a variational formulation of the robust function $\rho\left(\boldsymbol{f}; \epsilon, p\right)$ as the minimum of a quadratic function in $\boldsymbol{f}$ over the auxiliary variable $\boldsymbol{\beta}$.

Combining (6) and (14) we get:

$$\begin{aligned}\min_{\|\boldsymbol{f}\|_2=1} \rho\left(\mathbf{A}\boldsymbol{f}; \epsilon, p\right) = \min_{\|\boldsymbol{f}\|_2=1,\, \boldsymbol{\beta} \in \mathbb{R}_+^N} \phi\left(\boldsymbol{f}, \boldsymbol{\beta}\right) \equiv \\ \frac{p}{2}\boldsymbol{f}^{\mathsf{T}}\boldsymbol{\Gamma}\left(\boldsymbol{\beta}\right)\boldsymbol{f} + \sum_{n=1}^{N}\left(\frac{p\epsilon}{2}\beta_k^{p-2} + \frac{2-p}{2}\beta_k^p\right), \quad (15)\end{aligned}$$

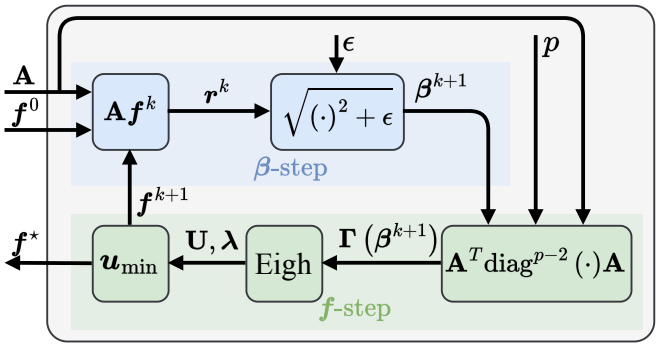

Fig. 2. The proposed IHLS layer architecture. $\boldsymbol{A}$, $\boldsymbol{f}^\star$ and $\{p, \epsilon\}$ are the input, output and parameters of the layer, respectively, while $\boldsymbol{f}^0$ serves as the initialization.

where

$$\boldsymbol{\Gamma}(\boldsymbol{\beta}) = \mathbf{A}^\top \mathrm{diag}^{p-2}(\boldsymbol{\beta})\,\mathbf{A} = \mathbf{A}^\top \mathbf{B} \mathbf{A}. \tag{16}$$

By exploiting the variational formulation of (14) we have achieved to express our original problem in (6) as a joint minimization in the variables $(\boldsymbol{f}, \boldsymbol{\beta})$, for which it is possible to consider the following alternating minimization strategy:

$$\boldsymbol{\beta} - \text{step} : \boldsymbol{\beta}^{k+1} = \underset{\boldsymbol{\beta} \in \mathbb{R}^N_+}{\arg\min}\, \phi\left(\boldsymbol{f}^k, \boldsymbol{\beta}\right), \tag{17a}$$

$$\boldsymbol{f} - \text{step} : \boldsymbol{f}^{k+1} = \underset{\|\boldsymbol{f}\|_2 = 1}{\arg\min}\, \phi\left(\boldsymbol{f}, \boldsymbol{\beta}^{k+1}\right). \tag{17b}$$

The minimizer of (17a) can be found by equating the gradient of the respective objective to zero, $\nabla_{\boldsymbol{\beta}} \phi\left(\boldsymbol{f}^k; \boldsymbol{\beta}\right) = \mathbf{0}$, which takes the form:

$$\kappa \left(\mathrm{diag}^2\left(\mathbf{A}\boldsymbol{f}^k\right) + \epsilon \boldsymbol{I} - \mathrm{diag}^2\left(\boldsymbol{\beta}\right)\right) \boldsymbol{\beta}^{p-3} = \mathbf{0}, \tag{18}$$

where $\kappa = p(p-2)/2$. For the case of interest $0 < p \leq 1$ and given that β_n is constrained to be strictly positive, the only admissible solution of Eq. (18) is

$$\beta_n^{k+1} = \sqrt{\left(\boldsymbol{e}_n^\top \mathbf{A} \boldsymbol{f}^k\right)^2 + \epsilon}, \;\; \forall n = 1, \ldots, N, \tag{19}$$

with $\boldsymbol{e}_n$ denoting the unit vector of the standard $\mathbb{R}^N$ basis.

Next, the minimizer of (17b) is the solution of a unit-norm constrained homogeneous least-squares problem [13], and is obtained as:

$$\boldsymbol{f}^{k+1} = \boldsymbol{u}_{\min}\left(\boldsymbol{\Gamma}\left(\boldsymbol{\beta}^{k+1}\right)\right), \tag{20}$$

where $\boldsymbol{u}_{\min}\left(\boldsymbol{\Gamma}\left(\boldsymbol{\beta}^{k+1}\right)\right)$ is the eigenvector associated to the smallest eigenvalue of the symmetric positive definite matrix $\boldsymbol{\Gamma}\left(\boldsymbol{\beta}^{k+1}\right)$.

Finally, by combining (16), (19), and (20), the overall minimization algorithm can be written in the compact form:

$$\boldsymbol{f}^{k+1} = \boldsymbol{u}_{\min}\left(\mathbf{A}^\top \mathrm{diag}^{\frac{p-2}{2}}\left(\left(\mathbf{A}\boldsymbol{f}^k\right)^2 + \epsilon \boldsymbol{I}\right) \mathbf{A}\right). \tag{21}$$

The derived update strategy corresponds to an iterative homogeneous least-squares solver (IHLS), where in each iteration k+1 the diagonal matrix $\mathbf{B}^k = \mathrm{diag}^{\frac{p-2}{2}}\left(\left(\mathbf{A}\boldsymbol{f}^k\right)^2 + \epsilon \boldsymbol{I}\right)$ is updated according to the solution of the previous iteration, $\boldsymbol{f}^k$. We provide a detailed schematic of IHLS in Fig. 2.

E. Supervision

We supervise two-view geometry estimation with $\mathcal{L}_{\text{twg}}$ that consists of two parts, pose supervision with $\mathcal{L}_{\text{pose}}$ and direct epipolar geometry supervision with $\mathcal{L}_{\text{epi}}$:

$$\mathcal{L}_{\text{twg}}(\mathbf{F}) = \mathcal{L}_{\text{pose}}(\hat{\mathbf{T}}, \mathbf{T}) + \lambda_{\text{epi}} \mathcal{L}_{\text{epi}}(\mathbf{F}, \boldsymbol{p}_{gt}), \tag{22}$$

where λ_{epi} is the balancing hyper-parameter.

The pose loss $\mathcal{L}_{\text{pose}}$ is computed between the ground-truth relative pose $\hat{\mathbf{T}} = \left[\hat{\mathbf{R}}, \hat{\mathbf{t}}\right]$ and the pose $\mathbf{T} = [\mathbf{R}, \mathbf{t}]$ extracted from the fundamental matrix given the ground-truth camera intrinsics. It is defined as the angular distance between the translation vectors $(\hat{\mathbf{t}}, \mathbf{t})$ and the angular distance between the rotation matrices $(\hat{\mathbf{R}}, \mathbf{R})$ balanced by the factor λ_{rot}:

$$\mathcal{L}_{\text{pose}}(\mathbf{F}) = \cos^{-1}\left(\frac{\hat{\mathbf{t}} \cdot \mathbf{t}}{\|\hat{\mathbf{t}}\|_2 \cdot \|\mathbf{t}\|_2}\right) \tag{23}$$

$$+ \lambda_{\text{rot}} \cos^{-1}\left(\frac{\mathrm{tr}\left(\hat{\mathbf{R}}^\top \mathbf{R}\right) - 1}{2}\right). \tag{24}$$

For the epipolar loss $\mathcal{L}_{\text{epi}}$ we use the *Sampson Epipolar Distance* [12] for the set of ground-truth matches $\boldsymbol{p}_{gt} = \{(\hat{\mathbf{x}}_v, \hat{\mathbf{x}}'_v)\}_{v=1}^V$ and the estimated fundamental matrix $\mathbf{F}$:

$$\mathcal{L}_{\text{epi}}(\mathbf{F}, \boldsymbol{p}_{gt}) = \frac{1}{V} \sum_{v=1}^{V} \frac{\hat{\mathbf{x}}_v'^T \mathbf{F} \hat{\mathbf{x}}_v}{\|\mathbf{F}\hat{\mathbf{x}}_v\|_2^2 + \|\mathbf{F}^T \hat{\mathbf{x}}'_v\|_2^2} \tag{25}$$

During the end-to-end training we also utilize $\mathcal{L}_{\text{match}}$ loss for the supervision of the feature extraction and matching stages. We refer to LoFTR [11] for the definition of $\mathcal{L}_{\text{match}}$.

The total loss $\mathcal{L}$ comprises the two-view geometry estimation loss $\mathcal{L}_{\text{twg}}$ for the matrices $\mathbf{F}_m$ from each iteration of our framework plus the feature loss $\mathcal{L}_{\text{match}}$ balanced by the factor λ:

$$\mathcal{L} = \frac{1}{M+1} \sum_{m=0}^{M} \mathcal{L}_{\text{twg}}(\mathbf{F}_m) + \lambda \mathcal{L}_{\text{match}}(\boldsymbol{I}_S, \boldsymbol{I}_T), \tag{26}$$

F. Training with Implicit Backpropagation

To use our robust function in an end-to-end network training pipeline, it is required that gradients can be back-propagated through the IHLS layer to its inputs and parameters, which we denote altogether as $\boldsymbol{\theta} = \{\boldsymbol{A}, p, \epsilon\}$. Specifically, we need to compute the gradients $\nabla_{\boldsymbol{\theta}} L = \nabla_{\boldsymbol{\theta}} \boldsymbol{f}^\star \nabla_{\boldsymbol{f}^\star} L$, where L is the final loss function and $\nabla_{\boldsymbol{f}^\star} L$ is computed according to the chain rule and its exact form depends on the operations involved in the succeeding layers of IHLS in the overall network.

We note, that $\boldsymbol{f}^\star$ corresponds to the minimizer of (6) and is attained through the iterative procedure outlined in (21), which is executed until convergence. A naïve approach would be to represent this iterative process as a recurrent computational graph, which can be unrolled using a pre-defined threshold on the total number of iterations. Backpropagation through time (BPTT) [44] or its truncated version (TBPTT) [45] could then be employed to perform the training. Such an approach can be rather inefficient

since it faces two important challenges, which can limit the maximum amount of utilized unrolled iterations and compromise the convergence of the optimization process. The first one is the prohibitively large amount of memory that might be required to store the intermediate results of each one of the unrolled iterations. The second reason is related to the instabilities arising from the limited precision of numerical calculations. This results in the vanishing/exploding gradients during training of recurrent architectures [46], and unstable backpropagation through eigendecomposition $\boldsymbol{u}_{\text{min}}$, which has to be performed for every iteration of (21).

To overcome all the above difficulties, we exploit the fact that any limit point of the sequence of solutions $\left\{\boldsymbol{f}^k\right\}$ produced by our proposed IHLS algorithm is a stationary point. This result is motivated by Proposition 2.7.1 in [47] and the fact that IHLS is a block coordinate descent minimization method, for which a unique solution exists for both subproblems in Eq. (17). Next, we utilize the Karush-Kuhn-Tucker (KKT) optimality condition [48] associated with the original problem in Eq. (6):

$$\begin{aligned}\nabla_{\boldsymbol{f}^\star}\left[\mathcal{L}\left(\boldsymbol{f}^\star,\lambda^\star\right)\equiv\rho\left(\mathbf{A}\boldsymbol{f}^\star;\epsilon,p\right)+\lambda^\star\left(\|\boldsymbol{f}^\star\|_2^2-1\right)\right]\\ =p\boldsymbol{\Gamma}\left(\boldsymbol{\beta}^\star\right)\boldsymbol{f}^\star+2\lambda^\star\boldsymbol{f}^\star=\mathbf{0},\end{aligned}\tag{27}$$

where $\mathcal{L}$ is the Lagrangian, λ^* denotes the Lagrange multiplier and $\boldsymbol{f}^\star$ represents the stationary point produced by IHLS. By multiplying both sides by $\boldsymbol{f}^{\star\mathsf{T}}$ and using that $\boldsymbol{f}^{\star\mathsf{T}}\boldsymbol{f}^\star=\|\boldsymbol{f}^\star\|_2^2=1$, we can compute the corresponding Lagrange multiplier $\lambda^\star$ as:

$$\lambda^\star=-\frac{p}{2}\boldsymbol{f}^{\star\mathsf{T}}\boldsymbol{\Gamma}\left(\boldsymbol{\beta}^\star\right)\boldsymbol{f}^\star.\tag{28}$$

Combining Eqs. (27) and (28) we end up with the following necessary condition that needs to be satisfied by the output of the IHLS layer upon convergence:

$$\boldsymbol{g}\left(\boldsymbol{f}^\star,\boldsymbol{\theta}\right)\equiv\left(\boldsymbol{I}-\boldsymbol{f}^\star\boldsymbol{f}^{\star\mathsf{T}}\right)\boldsymbol{\Gamma}\left(\boldsymbol{\beta}^\star\right)\boldsymbol{f}^\star=\mathbf{0},\tag{29}$$

where the dependence on $\boldsymbol{\theta}$ is hidden in both $\boldsymbol{f}^\star$ and $\boldsymbol{\beta}^\star$.

Now, if we differentiate both sides of this equation w.r.t $\boldsymbol{\theta}$, we obtain:

$$\begin{aligned}&\nabla_{\boldsymbol{\theta}}\boldsymbol{g}\left(\boldsymbol{f}^\star,\boldsymbol{\theta}\right)+\nabla_{\boldsymbol{\theta}}\boldsymbol{f}^\star\nabla_{\boldsymbol{f}^\star}\boldsymbol{g}\left(\boldsymbol{f}^\star,\boldsymbol{\theta}\right)=\mathbf{0}\\ &\Rightarrow\nabla_{\boldsymbol{\theta}}\boldsymbol{f}^\star=-\nabla_{\boldsymbol{\theta}}\boldsymbol{g}\left(\boldsymbol{f}^\star,\boldsymbol{\theta}\right)\left(\nabla_{\boldsymbol{f}^\star}\boldsymbol{g}\left(\boldsymbol{f}^\star,\boldsymbol{\theta}\right)\right)^{-1}.\end{aligned}\tag{30}$$

Therefore, it is possible to compute the gradient of the loss L w.r.t to the set of parameters $\boldsymbol{\theta}$ as:

$$\nabla_{\boldsymbol{\theta}}L=-\nabla_{\boldsymbol{\theta}}\boldsymbol{g}\left(\boldsymbol{f}^\star,\boldsymbol{\theta}\right)\left(\nabla_{\boldsymbol{f}^\star}\boldsymbol{g}\left(\boldsymbol{f}^\star,\boldsymbol{\theta}\right)\right)^{-1}\nabla_{\boldsymbol{f}^\star}L.\tag{31}$$

As a result of Eq. (31), backpropagation through our IHLS layer amounts to a single vector-jacobian product (VJP) $-\nabla_{\boldsymbol{\theta}}\boldsymbol{g}\left(\boldsymbol{f}^\star,\boldsymbol{\theta}\right)\boldsymbol{v}$, where the vector $\boldsymbol{v}$ is obtained as the solution of a system of linear equations $\nabla_{\boldsymbol{f}^\star}\boldsymbol{g}\left(\boldsymbol{f}^\star,\boldsymbol{\theta}\right)\boldsymbol{v}=\nabla_{\boldsymbol{f}^\star}L$, and $\nabla_{\boldsymbol{f}^\star}L$ is the incoming gradient. We note that in both cases where our IHLS layer is applied (fundamental or essential matrix estimation problems), $\nabla_{\boldsymbol{f}^\star}\boldsymbol{g}\left(\boldsymbol{f}^\star,\boldsymbol{\theta}\right)\in\mathbb{R}^{9\times9}$ corresponds to a relatively small matrix. Therefore, the computational burden introduced by the inversion of such a low-dimensional matrix is negligible. Furthermore, our derived procedure is agnostic to the number of iterations performed during the forward pass, and requires only a single backpropagation through $\boldsymbol{u}_{\text{min}}$ to be performed in $-\nabla_{\boldsymbol{\theta}}\boldsymbol{g}\left(\boldsymbol{f}^\star,\boldsymbol{\theta}\right)\boldsymbol{v}$, so it addresses all drawbacks of BPTT that we highlighted earlier. We provide a detailed implementation-friendly summary of our backpropagation procedure in Algorithm 1.

Algorithm 1: Backward pass through the IHLS layer

Inputs: $\nabla_{\boldsymbol{f}^\star}L$ `/* the incoming gradient */`
Buffer from Forward pass: $\boldsymbol{f}^\star$, $\boldsymbol{\theta}=\{\mathbf{A},p,\epsilon\}$
Result: $\nabla_{\boldsymbol{\theta}}L$
Record computational graph with autograd
- $\boldsymbol{r}^\star\leftarrow\mathbf{A}\boldsymbol{f}^\star$
- $\boldsymbol{\beta}^\star\leftarrow\sqrt{\left(\boldsymbol{r}^\star\right)^2+\epsilon}$ `/* square and square root are applied element-wise */`
- $\boldsymbol{\Gamma}\left(\boldsymbol{\beta}^\star\right)\leftarrow\mathbf{A}^{\mathsf{T}}\text{diag}^{p-2}\left(\boldsymbol{\beta}^\star\right)\mathbf{A}$
- $\boldsymbol{g}\left(\boldsymbol{f}^\star,\boldsymbol{\theta}\right)\leftarrow\boldsymbol{\Gamma}\left(\boldsymbol{\beta}^\star\right)\boldsymbol{f}^\star-\left(\boldsymbol{f}^{\star\mathsf{T}}\boldsymbol{\Gamma}\left(\boldsymbol{\beta}^\star\right)\boldsymbol{f}^\star\right)\boldsymbol{f}^\star$

$\mathbf{B}\leftarrow\frac{\partial\boldsymbol{g}}{\partial\boldsymbol{f}^\star}$ `/* Jacobian computed by autograd */`
$\boldsymbol{v}\leftarrow\mathbf{B}^{-\mathsf{T}}\nabla_{\boldsymbol{f}^\star}L$
$\nabla_{\boldsymbol{\theta}}L\leftarrow\boldsymbol{v}^{\mathsf{T}}\frac{\partial\boldsymbol{g}}{\partial\boldsymbol{\theta}}$ `/* VJP computed by autograd */`

We note, that in Eq. (30) of our derivations we utilize the implicit function theorem, so our backpropagation strategy belongs to the emerging class of the implicit differentiation approaches [49]–[51]. We find our training strategy to be similar to one proposed in [52], where the implicit backpropagation rules were derived for several unconstrained minimization problems. However, this approach is not directly applicable to our case, since in Eq. (6) we are dealing with a constrained minimization problem.

IV. EXPERIMENTS

We evaluate the performance of the proposed approach on the outdoor and indoor relative camera pose estimation tasks. We use ScanNet [53] for the indoor scenario and Megadepth [54] for outdoor task. As the evaluation measure we use AUC of the pose error at thresholds $(5^\circ,10^\circ,20^\circ)$ following [10], [11].

To perform end-to-end training of the full pipeline in the same setup that was used for training of the individual parts, we first train LoFTR [11] from scratch following the authors' strategy. We then freeze the feature extraction and matching stages and train the two-view geometry estimation module alone. Finally, we perform end-to-end training of the feature extraction, matching and two-view geometry estimation. The training and evaluation indices both for outdoor and indoor datasets are the same as in [11].

All the experiments are conducted on the computational cluster of 8 NVIDIA V100 GPU's. The values of the hyperparameters are the same for both scenarios $\lambda=1$, $\lambda_{rot}=10$ and $\lambda_{epi}=1\times10^{-3}$. We use Adam optimizer with the initial learning rate of 1×10^{-4} for the two-view geometry estimation training and 1×10^{-5} for the end-to-end setup.

A. Baseline methods

We perform comparison with both classic and learning-based solutions. First, we compare with RANSAC [17] robust estimator, which is a standard and the most widely used algorithm for the task. We also compare with the current state-of-the-art approach MAGSAC++ [22]. Among

TABLE I

EVALUATION ON MEGADEPTH [54] FOR OUTDOOR POSE ESTIMATION

Category	Method	Pose error AUC (%) ↑		
		@5°	@10°	@20°
classic	RANSAC [17]	31.53	46.71	61.38
	MAGSAC++ [22]	39.15	53.78	67.00
learning-based	DFE (5 iterations) [29]	53.08	66.89	77.7
	∇-RANSAC [36]	47.83	61.98	73.34
	Ours	54.86	68.46	78.94
	Ours (end2end)	**56.63**	**70.02**	**80.48**

TABLE II

EVALUATION ON SCANNET [53] FOR INDOOR POSE ESTIMATION

Category	Method	Pose error AUC (%) ↑		
		@5°	@10°	@20°
classic	RANSAC [17]	09.76	22.38	37.04
	MAGSAC++ [22]	13.96	27.88	42.39
learning-based	DFE (5 iterations) [29]	18.02	34.76	51.33
	∇-RANSAC [36]	15.24	29.17	45.38
	Ours	19.19	36.22	52.67
	Ours (end2end)	**21.99**	**39.70**	**56.13**

learning-based solutions we choose the robust loss function fitting approach DFE [29] that implicitly learns an optimal robust loss function, as this method belongs to the same category as ours. Last, we compare with the recent method ∇-RANSAC [36] that improves matching confidences estimated with CLNet [55] by backpropagation of the learning signals through the hypotheses sampling. For the fair comparison, we trained CLNet+∇-RANSAC in the same setup as ours, with the same input $(\boldsymbol{p}, \boldsymbol{\tau})$ to the network and the same loss (22). During inference of ∇-RANSAC, the CLNet [55] was evaluated in combination with MAGSAC++ and Plackett-Luce model sampling following authors [36].

For reasons of clarity, where possible we perform hyper-parameter tuning for all the baseline methods, e.g. in MAGSAC++ we increased the default sigma threshold from 1 to 5. The learning-based methods are trained in the same setup as ours.

B. Outdoor camera pose estimation

In the outdoor scenario we use MegaDepth [54] dataset that consists of 1M images of 196 outdoor scenes with challenging repetitive patterns and natural illumination changes. For training and evaluation the images are resized to the 840 pixels across the biggest dimension. The two-view geometry estimation is trained for 48 epochs with batch size 16 and end-to-end pipeline for 16 epochs with batch size 8. For the evaluation we use the same image pairs as in [11].

As can be seen in Table I, our method outperforms classic approach MAGSAC++ [22] by a large margin (40% at AUC@5°) and performs 3 % better at AUC@5° than the learning-based competitor DFE [29]. Moreover, end-to-end training with matching stage further improves the results.

C. Indoor camera pose estimation

For the evaluation in the indoor environment we use ScanNet [53] dataset that contains 1.5k sequences and exhibits images with low textures and challenging lighting conditions. Compared to MegaDepth [54], the performance of existing approaches on ScanNet [53] is worse as the outlier rate is higher. In our experiments the images and the corresponding depth maps are resized to the resolution of 640×480. We train two-view geometry estimation for 64 epochs with batch size 32 and end-to-end pipeline for 16 epochs with batch size 16. During evaluation we use the same image pairs as in [11].

Table II demonstrates that our method outperforms both classic and learning-based approaches. We achieve approximately the same supremacy over the classic methods as in the outdoor scenario (37% at AUC@5°). And more importantly, compared to the outdoor, we achieve bigger improvement over the learning-based approaches (6% at AUC@5° for DFE [29]). This indicates that our method handles better the cases where the outlier rate is higher.

V. CONCLUSIONS

In this paper we have presented a method for outlier-robust two-view geometry estimation. Based on the feature correspondences, we first estimate the initial solution and then iteratively refine it with our outlier-robust IHLS solver. Our experiments show that the proposed approach achieves state-of-the-art results on the relative camera pose estimation task in both indoor and oudoor environments.

VI. ACKNOWLEDGEMENTS

The work was supported by the Analytical center under the RF Government (subsidy agreement 000000D730321P5Q0002, Grant No. 70-2021-00145 02.11.2021)

REFERENCES

[1] J. Wang, Y. Zhong, Y. Dai, S. Birchfield, K. Zhang, N. Smolyanskiy, and H. Li, "Deep two-view structure-from-motion revisited," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 8953–8962.

[2] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang, L. Lu, X. Jia, Q. Liu, J. Dai, Y. Qiao, and H. Li, "Planning-oriented autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 17 853–17 862.

[3] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *ECCV*, 2020.

[4] Y.-Y. Jau, R. Zhu, H. Su, and M. Chandraker, "Deep keypoint-based camera pose estimation with geometric constraints," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 4950–4957.

[5] C. M. Parameshwara, G. Hari, C. Fermüller, N. J. Sanket, and Y. Aloimonos, "Diffposenet: Direct differentiable camera pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 6845–6854.

[6] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, pp. 91–110, 2004.

[7] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "Lift: Learned invariant feature transform," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 467–483.

[8] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 337–33 712, 2017.

[9] M. Tyszkiewicz, P. Fua, and E. Trulls, "Disk: Learning local features with policy gradient," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 14 254–14 265.

[10] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[11] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, "Loftr: Detector-free local feature matching with transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 8922–8931.

[12] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[13] R. Szeliski, *Computer vision: algorithms and applications*. Springer Nature, 2022.

[14] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment - a modern synthesis," in *Workshop on Vision Algorithms*, 1999.

[15] H. C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, vol. 293, pp. 133–135, 1981.

[16] O. D. Faugeras, "What can be seen in three dimensions with an uncalibrated stereo rig," in *European Conference on Computer Vision*, 1992.

[17] 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*, vol. 24, no. 6, p. 381–395, jun 1981. [Online]. Available: https://doi.org/10.1145/358669.358692

[18] P. Torr and A. Zisserman, "Mlesac: A new robust estimator with application to estimating image geometry," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 138–156, 2000.

[19] O. Chum, J. Matas, and J. Kittler, "Locally optimized ransac," in *DAGM-Symposium*, 2003.

[20] O. Chum, T. Werner, and J. Matas, "Two-view geometry estimation unaffected by a dominant plane," *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 772–779 vol. 1, 2005.

[21] D. Barath, J. Matas, and J. Noskova, "Magsac: Marginalizing sample consensus," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[22] D. Barath, J. Noskova, M. Ivashechkin, and J. Matas, "Magsac++, a fast, reliable and accurate robust estimator," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[23] R. Hartley, "Projective reconstruction and invariants from multiple images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 10, pp. 1036–1041, 1994.

[24] D. Nister, "An efficient solution to the five-point relative pose problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–770, 2004.

[25] P. H. S. Torr and D. W. Murray, "The development and comparison of robust methods for estimating the fundamental matrix," *International Journal of Computer Vision*, vol. 24, pp. 271–300, 1997.

[26] A. Ruckstuhl, "Robust fitting of parametric models based on m-estimation," *Lecture notes*, p. 40, 2014.

[27] P. J. Huber, "Robust estimation of a location parameter," *Annals of Mathematical Statistics*, vol. 35, pp. 492–518, 1964.

[28] L. Peng, C. Kümmerle, and R. Vidal, "On the convergence of irls and its variants in outlier-robust estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 17 808–17 818.

[29] R. Ranftl and V. Koltun, "Deep fundamental matrix estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[30] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International Conference on Computer Vision*, 2011, pp. 2564–2571.

[31] P. Lindenberger, P.-E. Sarlin, and M. Pollefeys, "LightGlue: Local Feature Matching at Light Speed," in *ICCV*, 2023.

[32] J. Yu, J. Chang, J. He, T. Zhang, J. Yu, and F. Wu, "Adaptive spot-guided transformer for consistent local feature matching," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 21 898–21 908.

[33] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother, "Dsac - differentiable ransac for camera localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[34] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Mach. Learn.*, vol. 8, no. 3–4, p. 229–256, may 1992. [Online]. Available: https://doi.org/10.1007/BF00992696

[35] E. Brachmann and C. Rother, "Neural-guided ransac: Learning where to sample model hypotheses," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[36] T. Wei, Y. Patel, A. Shekhovtsov, J. Matas, and D. Barath, "Generalized differentiable ransac," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 17 649–17 660.

[37] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," in *International Conference on Learning Representations*, 2017.

[38] K. MacTavish and T. D. Barfoot, "At all costs: A comparison of robust cost functions for camera correspondence outliers," *2015 12th Conference on Computer and Robot Vision*, pp. 62–69, 2015.

[39] D. Ikami, T. Yamasaki, and K. Aizawa, "Fast and robust estimation for unit-norm constrained linear fitting problems," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[40] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[41] D. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[42] R. Chartrand, "Exact reconstruction of sparse signals via nonconvex minimization," *IEEE Signal Processing Letters*, vol. 14, no. 10, pp. 707–710, 2007.

[43] Y. Sun, P. Babu, and D. P. Palomar, "Majorization-minimization algorithms in signal processing, communications, and machine learning," *IEEE Transactions on Signal Processing*, vol. 65, no. 3, pp. 794–816, 2017.

[44] A. Robinson and F. Fallside, *The utility driven dynamic error propagation network*. University of Cambridge Department of Engineering Cambridge, 1987.

[45] F. Kokkinos and S. Lefkimmiatis, "Iterative joint image demosaicking and denoising using a residual denoising network," *IEEE Transactions on Image Processing*, vol. 28, no. 8, pp. 4177–4188, 2019.

[46] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International conference on machine learning*. PMLR, 2013, pp. 1310–1318.

[47] D. Bertsekas, *Constrained optimization and Lagrange multiplier methods*. Athena Scientific, 1996.

[48] S. Boyd and L. Vandenberghe, *Convex Optimization*. Kluwer Academic Publishers, 2004.

[49] S. Bai, J. Z. Kolter, and V. Koltun, "Deep equilibrium models," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[50] N. Zucchet and J. Sacramento, "Beyond Backpropagation: Bilevel Optimization Through Implicit Differentiation and Equilibrium Propagation," *Neural Computation*, vol. 34, no. 12, pp. 2309–2346, 11 2022.

[51] S. Gould, R. Hartley, and D. Campbell, "Deep declarative networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 08, pp. 3988–4004, aug 2022.

[52] S. Lefkimmiatis and I. S. Koshelev, "Learning sparse and low-rank priors for image recovery via iterative reweighted least squares minimization," in *The Eleventh International Conference on Learning Representations*, 2023.

[53] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[54] Z. Li and N. Snavely, "Megadepth: Learning single-view depth prediction from internet photos," in *Computer Vision and Pattern Recognition (CVPR)*, 2018.

[55] C. Zhao, Y. Ge, F. Zhu, R. Zhao, H. Li, and M. Salzmann, "Progressive correspondence pruning by consensus learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 6464–6473.