RoMa: Robust Dense Feature Matching

Johan Edstedt[1]   Qiyu Sun[2]   Georg Bökman[3]   Mårten Wadenbäck[1]   Michael Felsberg[1]

[1]Linköping University, [2]East China University of Science and Technology, [3]Chalmers University of Technology

Figure 1. **RoMa is robust, *i.e.*, able to match under extreme changes.** We propose RoMa, a model for dense feature matching that is robust to a wide variety of challenging real-world changes in scale, illumination, viewpoint, and texture. We show correspondences estimated by RoMa on the extremely challenging benchmark WxBS [37], where most previous methods fail, and on which we set a new state-of-the-art with an improvement of **36% mAA**. The estimated correspondences are visualized by grid sampling coordinates bilinearly from the other image, using the estimated warp, and multiplying with the estimated confidence.

Abstract

Feature matching is an important computer vision task that involves estimating correspondences between two images of a 3D scene, and dense methods estimate all such correspondences. The aim is to learn a robust model, i.e., a model able to match under challenging real-world changes. In this work, we propose such a model, leveraging frozen pretrained features from the foundation model DINOv2. Although these features are significantly more robust than local features trained from scratch, they are inherently coarse. We therefore combine them with specialized ConvNet fine features, creating a precisely localizable feature pyramid. To further improve robustness, we propose a tailored transformer match decoder that predicts anchor probabilities, which enables it to express multimodality. Finally, we propose an improved loss formulation through regression-by-classification with subsequent robust regression. We conduct a comprehensive set of experiments that show that our method, RoMa, achieves significant gains, setting a new state-of-the-art. In particular, we achieve a 36% improvement on the extremely challenging WxBS benchmark. Code is provided at github.com/Parskatt/RoMa.

1. Introduction

Feature matching is the computer vision task of from two images estimating pixel pairs that correspond to the same 3D point. It is crucial for downstream tasks such as 3D reconstruction [47] and visual localization [43]. Dense feature matching methods [17, 38, 53, 56] aim to find all matching pixel-pairs between the images. These dense methods employ a coarse-to-fine approach, whereby matches are first predicted at a coarse level and successively refined at finer resolutions. Previous methods commonly learn coarse features using 3D supervision [17, 44, 48, 56]. While this allows for specialized coarse features, it comes with downsides. In particular, since collecting real-world 3D datasets is expensive, the amount of available data is limited, which means models risk overfitting to the training set. This in turn limits the models robustness to scenes that differ significantly from what has been seen during training. A well-known approach to limit overfitting is to freeze the backbone used [31, 51, 58]. However, using frozen backbones pretrained on ImageNet classification, the out-of-the-box performance is insufficient for feature matching (see experiments in Table 1). A recent promising direction for frozen pretrained features is large-scale self-supervised pretraining using Masked image Modeling (MIM) [25, 39, 60, 66]. The methods, including DI-
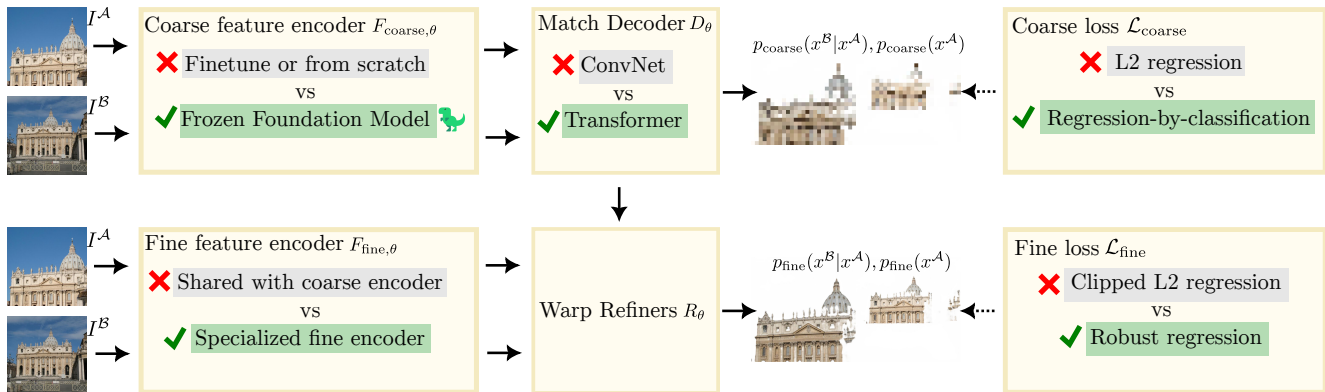
Figure 2. **Illustration of our robust approach RoMa.** Our contributions are shown with green highlighting and a checkmark, while previous approaches are indicated with gray highlights and a cross. Our first contribution is using a frozen foundation model for coarse features, compared to fine-tuning or training from scratch. DINOv2 lacks fine features, which are needed for accurate correspondences. To tackle this, we combine the DINOv2 coarse features with specialized fine features from a ConvNet, see Section 3.2. Second, we propose an improved coarse match decoder D_θ, which typically is a ConvNet, with a coordinate agnostic Transformer decoder that predicts anchor probabilities instead of directly regressing coordinates, see Section 3.3. Third, we revisit the loss functions used for dense feature matching. We argue from a theoretical model that the global matching stage needs to model multimodal distributions, and hence use a regression-by-classification loss instead of an L2 loss. For the refinement, we in contrast use a robust regression loss, as the matching distribution is locally unimodal. These losses are further discussed in Section 3.4. The impact of our contributions is ablated in our extensive ablation study in Table 2.

NOv2 [64], retain local information better than classification pretraining [64] and have been shown to generate features that generalize well to dense vision tasks. However, the application of DINOv2 in dense feature matching is still complicated due to the lack of fine features, which are needed for refinement.

We overcome this issue by leveraging a frozen DINOv2 encoder for coarse features, while using a proposed specialized ConvNet encoder for the fine features. This has the benefit of incorporating the excellent general features from DINOv2, while simultaneuously having highly precise fine features. We find that features specialized for only coarse matching or refinement significantly outperform features trained for both tasks jointly. These contributions are presented in more detail in Section 3.2. We additionally propose a Transformer match decoder that while also increasing performance for the baseline, particularly improves performance when used to predict anchor probabilities instead of regressing coordinates in conjunction with the DINOv2 coarse encoder. This contribution is elaborated further in Section 3.3.

Lastly, we investigate how to best train dense feature matchers. Recent SotA dense methods such as DKM [17] use a non-robust regression loss for the coarse matching as well as for the refinement. We argue that this is not optimal as the matching distribution at the coarse stage is often multimodal, while the conditional refinement is more likely to be unimodal. Hence requiring different approaches to training. We motivate this from a theoretical framework

in Section 3.4. Our framework motivates a division of the coarse and fine losses into seperate paradigms, regression-by-classification for the global matches using coarse features, and robust regression for the refinement using fine features.

Our full approach, which we call RoMa, is robust to extremely challenging real-world cases, as we demonstrate in Figure 1. We illustrate our approach schematically in Figure 2. In summary, our contributions are as follows:

(a) We integrate frozen features from the foundation model DINOv2 [39] for dense feature matching. We combine the coarse features from DINOv2 with specialized fine features from a ConvNet to produce a precisely localizable yet robust feature pyramid. See Section 3.2.

(b) We propose a Transformer-based match decoder, which predicts anchor probabilities instead of coordinates. See Section 3.3.

(c) We improve the loss formulation. In particular, we use a regression-by-classification loss for coarse global matches, while we use robust regression loss for the refinement stage, both of which we motivate from a theoretical analysis. See Section 3.4.

(d) We conduct an extensive ablation study over our contributions, and SotA experiments on a set of diverse and competitive benchmarks, and find that RoMa sets a new state-of-the-art. In particular, achieving a gain of 36% on the difficult WxBS benchmark. See Section 4.

2. Related Work

2.1. Sparse → Detector Free → Dense Matching

Feature matching has traditionally been approached by keypoint detection and description followed by matching the descriptions [4, 10, 15, 18, 35, 41, 44, 57]. Recently, the detector-free approach [9, 13, 48, 50] replaces the keypoint detection with dense matching on a coarse scale, followed by mutual nearest neighbors extraction, which is followed by refinement. The dense approach [17, 36, 38, 54, 55, 67] instead estimates a dense warp, aiming to estimate every matchable pixel pair.

2.2. Self-Supervised Vision Models

Foundation models [7] pre-trained on large quantities of data have recently demonstrated significant potential in learning all-purpose features for various visual models via self-supervised learning. Caron et al. [12] observe that self-supervised ViT features capture more distinct information than supervised models do. iBOT [66] explores MIM within a self-distillation framework to develop a semantically rich visual tokenizer. DINOv2 [39] reveals that self-supervised methods can produce all-purpose visual features.

2.3. Frozen Features for Matching

Typically, detector-free SotA matching fine-tune [17] or train the backbone from scratch [48]. There are however some exceptions [28, 42] that use frozen ImageNet features. As we demonstrate in Section 3.2 such features are not sufficiently robust, hence necessitating our foundation model approach. Detector-based matchers [33, 44] use frozen descriptors as input. However, the motivation is to ensure compatibility with the descriptor, and to the best of our knowledge it has not been investigated whether fine-tuning/training end-to-end would improve performance.

2.4. Robust Loss Formulations

Robust Regression Losses: Robust loss functions provide a continuous transition between an inlier distribution , and an outlier distribution. Robust losses have been used as regularizers for optical flow [5, 6], robust smoothing [19], and as loss functions [3, 34].

Regression by Classification: Regression by classification [52, 61, 62] involves casting regression problems as classification. This is particularly useful for sharp borders in motion, such as stereo disparity [20, 23], and has also been used for absolute pose regression [21].

Classification then Regression: Li et al. [29], and Budvytis et al. [8] proposed hierarchical classification-regression frameworks for visual localization. Sun et al. [48] optimize the model log-likelihood of mutual nearest neighbors, followed by L2 regression-based refinement for feature matching.

3. Method

In this section, we detail our method. We begin with preliminaries and notation for dense feature matching in Section 3.1. We then discuss our incorporation of DINOv2 [39] as a coarse encoder, and specialized fine features in Section 3.2. We present our proposed Transformer match decoder in Section 3.3. Finally, our proposed loss formulation in Section 3.4. A summary and visualization of our full approach is provided in Figure 2. Further details on the exact architecture are given in the supplementary.

3.1. Preliminaries on Dense Feature Matching

Dense feature matching is, given two images $I^{\mathcal{A}}, I^{\mathcal{B}}$, to estimate a dense warp $W^{\mathcal{A} \to \mathcal{B}}$ (mapping coordinates $x^{\mathcal{A}}$ from $I^{\mathcal{A}}$ to $x^{\mathcal{B}}$ in $I^{\mathcal{B}}$), and a matchability score $p(x^{\mathcal{A}})$[1] for each pixel. From a probabilistic perspective, $p(W^{\mathcal{A} \to \mathcal{B}}) = p(x^{\mathcal{B}}|x^{\mathcal{A}})$ is the conditional matching distribution. Multiplying $p(x^{\mathcal{B}}|x^{\mathcal{A}})p(x^{\mathcal{A}})$ yields the joint distribution. We denote the model distribution as $p_\theta(x^{\mathcal{A}}, x^{\mathcal{B}}) = p_\theta(x^{\mathcal{B}}|x^{\mathcal{A}})p_\theta(x^{\mathcal{A}})$. When working with warps, *i.e.*, where $p_\theta(x^{\mathcal{B}}|x^{\mathcal{A}})$ has been converted to a deterministic mapping, we denote the model warp as $\hat{W}^{\mathcal{A} \to \mathcal{B}}$. Viewing the predictive distribution as a warp is natural in high resolution, as it can then be seen as a deterministic mapping. However, due to multimodality, it is more natural to view it in the probabilistic sense at coarse scales.

The end goal is to obtain a good estimate over correspondences of coordinates $x^{\mathcal{A}}$ in image $I^{\mathcal{A}}$ and coordinates $x^{\mathcal{B}}$ in image $I^{\mathcal{B}}$. For dense feature matchers, estimation of these correspondences is typically done by a one-shot coarse *global matching* stage (using coarse features) followed by subsequent *refinement* of the estimated warp and confidence (using fine features).

We use the recent SotA dense feature matching model DKM [17] as our baseline. For consistency, we adapt the terminology used there. We denote the coarse features used to estimate the initial warp, and the fine features used to refine the warp by

$$\{\varphi^{\mathcal{A}}_{\text{coarse}}, \varphi^{\mathcal{A}}_{\text{fine}}\} = F_\theta(I^{\mathcal{A}}), \{\varphi^{\mathcal{B}}_{\text{coarse}}, \varphi^{\mathcal{B}}_{\text{fine}}\} = F_\theta(I^{\mathcal{B}}), \quad (1)$$

where F_θ is a neural network feature encoder. We will leverage DINOv2 for extraction of $\varphi^{\mathcal{A}}_{\text{coarse}}, \varphi^{\mathcal{B}}_{\text{coarse}}$, however, DINOv2 features are not precisely localizable, which we tackle by combining the coarse features with precise local features from a specialized ConvNet backbone. See Section 3.2 for details.

The coarse features are matched with global matcher G_θ consisting of a match encoder E_θ and match decoder D_θ,

$$\begin{cases} (\hat{W}^{\mathcal{A} \to \mathcal{B}}_{\text{coarse}}, \ p^{\mathcal{A}}_{\theta, \text{coarse}}) = G_\theta(\varphi^{\mathcal{A}}_{\text{coarse}}, \varphi^{\mathcal{B}}_{\text{coarse}}), \\ G_\theta(\varphi^{\mathcal{A}}_{\text{coarse}}, \varphi^{\mathcal{B}}_{\text{coarse}}) = D_\theta(E_\theta(\varphi^{\mathcal{A}}_{\text{coarse}}, \varphi^{\mathcal{B}}_{\text{coarse}})). \end{cases} \quad (2)$$

---

[1]This is denoted as $p^{\mathcal{A} \to \mathcal{B}}$ by Edstedt et al. [17]. We omit the $\mathcal{B}$ to avoid confusion with the conditional.

We use a Gaussian Process [40] as the match encoder E_θ as in previous work [17]. However, while our baseline uses a ConvNet to decode the matches, we propose a Transformer match decoder D_θ that predicts anchor probabilities instead of directly regressing the warp. This match decoder is particularly beneficial in our final approach (see Table 2). We describe our proposed match decoder in Section 3.3. The refinement of the coarse warp $\hat{W}^{\mathcal{A}\to\mathcal{B}}_{\text{coarse}}$ is done by the refiners R_θ,

$$\left(\hat{\mathbf{w}}^{\mathcal{A}\to\mathcal{B}}, p^{\mathcal{A}}_\theta\right) = R_\theta\left(\varphi^{\mathcal{A}}_{\text{fine}}, \varphi^{\mathcal{B}}_{\text{fine}}, \hat{\mathbf{W}}^{\mathcal{A}\to\mathcal{B}}_{\text{coarse}}, p^{\mathcal{A}}_{\theta,\text{coarse}}\right). \quad (3)$$

As in previous work, the refiner is composed of a sequence of ConvNets (using strides $\{1, 2, 4, 8\}$) and can be decomposed recursively as

$$\left(\hat{W}^{\mathcal{A}\to\mathcal{B}}_i, p^{\mathcal{A}}_{i,\theta}\right) = R_{\theta,i}(\varphi^{\mathcal{A}}_i, \varphi^{\mathcal{B}}_i, \hat{W}^{\mathcal{A}\to\mathcal{B}}_{i+1}, p^{\mathcal{A}}_{\theta,i+1}), \quad (4)$$

where the stride is 2^i. The refiners predict a residual offset for the estimated warp, and a logit offset for the certainty. As in the baseline they are conditioned on the outputs of the previous refiner by using the previously estimated warp to a) stack feature maps from the images, and b) construct a local correlation volume around the previous target.

The process is repeated until reaching full resolution. We use the same architecture as in the baseline. Following DKM, we detach the gradients between the refiners and upsample the warp bilinearly to match the resolution of the finer stride.

Probabilistic Notation: When later defining our loss functions, it will be convenient to refer to the outputs of the different modules in a probabilistic notation. We therefore introduce this notation here first for clarity.

We denote the probability distribution modeled by the global matcher as

$$p_\theta(x^{\mathcal{A}}_{\text{coarse}}, x^{\mathcal{B}}_{\text{coarse}}) = G_\theta(\varphi^{\mathcal{A}}_{\text{coarse}}, \varphi^{\mathcal{B}}_{\text{coarse}}). \quad (5)$$

Here we have dropped the explicit dependency on the features and the previous estimate of the marginal for notational brevity. Note that the output of the global matcher will sometimes be considered as a discretized distribution using anchors, or as a decoded warp. We do not use separate notation for these two different cases to keep the notation uncluttered.

We denote the probability distribution modeled by a refiner at scale $s = c2^i$ as

$$p_\theta(x^{\mathcal{A}}_i, x^{\mathcal{B}}_i | \hat{W}^{\mathcal{A}\to\mathcal{B}}_{i+1}) = R_{\theta,i}(\varphi^{\mathcal{A}}_i, \varphi^{\mathcal{B}}_i, \hat{W}^{\mathcal{A}\to\mathcal{B}}_{i+1}, p^{\mathcal{A}}_{\theta,i+1}), \quad (6)$$

The basecase $\hat{W}^{\mathcal{A}\to\mathcal{B}}_{\text{coarse}}$ is computed by decoding $p_\theta(x^{\mathcal{B}}_{\text{coarse}} | x^{\mathcal{A}}_{\text{coarse}})$. As for the global matcher we drop the explicit dependency on the features.

Table 1. **Evaluation of frozen features on MegaDepth.** We compare the VGG19 and ResNet50 backbones commonly used in feature matching with the generalist features of DINOv2.

Method	EPE $\downarrow$	Robustness % $\uparrow$
VGG19	87.6	43.2
RN50	60.2	57.5
DINOv2	**27.1**	**85.6**

3.2. Robust and Localizable Features

We first investigate the robustness of DINOv2 to viewpoint and illumination changes compared to VGG19 and ResNet50 on the MegaDepth [30] dataset. To decouple the backbone from the matching model we train a single linear layer on top of the frozen model followed by a kernel nearest neighbour matcher for each method. We measure the performance both in average end-point-error (EPE) on a standardized resolution of 448×448, and by what we call the Robustness % which we define as the percentage of matches with an error lower than 32 pixels. We refer to this as robustness, as, while these matches are not necessarily precise, it is typically sufficient for the refinement stage to produce a correct adjustment.

We present results in Table 1. We find that DINOv2 features are significantly more robust to changes in viewpoint than both ResNet and VGG19. Interestingly, we find that the VGG19 features are worse than the ResNet features for coarse matching, despite VGG features being widely used as local features [16, 45, 56]. Further details of this experiment are provided in the supplementary material.

In DKM [17], the feature encoder F_θ is assumed to consist of a single network producing a feature pyramid of coarse and fine features used for global matching and refinement respectively. This is problematic when using DINOv2 features as only features of stride 14 exist. We therefore decouple F_θ into $\{F_{\text{coarse},\theta}, F_{\text{fine},\theta}\}$ and set $F_{\text{coarse},\theta} = $ DINOv2. The coarse features are extracted as

$$\varphi^{\mathcal{A}}_{\text{coarse}} = F_{\text{coarse},\theta}(I^{\mathcal{A}}), \varphi^{\mathcal{B}}_{\text{coarse}} = F_{\text{coarse},\theta}(I^{\mathcal{B}}). \quad (7)$$

We keep the DINOv2 encoder frozen throughout training. This has two benefits. The main benefit is that keeping the representations fixed reduces overfitting to the training set, enabling RoMa to be more robust. It is also additionally significantly cheaper computationally and requires less memory. However, DINOv2 cannot provide fine features. Hence a choice of $F_{\text{fine},\theta}$ is needed. While the same encoder for fine features as in DKM could be chosen, *i.e.*, a ResNet50 (RN50) [24], it turns out that this is not optimal.

We begin by investigating what happens by simply decoupling the coarse and fine feature encoder, *i.e.*, not sharing weights between the coarse and fine encoder (even when using the same network). We find that, as supported by

Setup II in Table 2, this significantly increases performance. This is due to the feature extractor being able to specialize in the respective tasks, and we hence call this *specialization*.

This raises a question, VGG19 features, while less suited for coarse matching (see Table 1), could be better suited for fine localized features. We investigate this by setting $F_{\text{fine},\theta} = \text{VGG19}$ in Setup III in Table 2. Interestingly, even though VGG19 coarse features are significantly worse than RN50, we find that they significantly outperform the RN50 features when leveraged as fine features. Our finding indicates that there is an inherent tension between fine localizability and coarse robustness. We thus use VGG19 fine features in our full approach.

3.3. Transformer Match Decoder D_θ

Regression-by-Classification: We propose to use the *regression by classification* formulation for the match decoder, whereby we discretize the output space. We choose the following formulation,

$$p_{\text{coarse},\theta}(x^{\mathcal{B}}|x^{\mathcal{A}}) = \sum_{k=1}^{K} \pi_k(x^{\mathcal{A}})\mathcal{B}_{m_k}, \qquad (8)$$

where K is the quantization level, π_k are the probabilities for each component, $\mathcal{B}$ is some 2D base distribution, and $\{m_k\}_1^K$ are anchor coordinates. In practice, we use $K = 64 \times 64$ classification anchors positioned uniformly as a tight cover of the image grid, and $\mathcal{B} = \mathcal{U}$, *i.e.*, a uniform distribution[2]. Note that base distribution $\mathcal{B}_{m_k}$ is non-zero only in a square area of size $\frac{2}{K} \times \frac{2}{K}$ on the normalized image grid around its anchor m_k and 0 outside, and therefore $p(x^{\mathcal{B}}|x^{\mathcal{A}}) \neq 1$ in general. We denote the probability of an anchor as π_k and the coordinate on the grid as m_k.

For refinement, the conditional is converted to a deterministic warp per pixel. We decode the warp by argmax over the classification anchors, $k^*(x) = \arg\max_k \pi_k(x)$, followed by a local adjustment which can be seen as a local softargmax. Mathematically,

$$\text{ToWarp}(p_{\text{coarse},\theta}(x_{\text{coarse}}^{\mathcal{B}}|x_{\text{coarse}}^{\mathcal{A}})) =$$

$$\frac{\sum_{i \in N_4(k^*(x_{\text{coarse}}^{\mathcal{A}}))} \pi_i m_i}{\sum_{i \in N_4(k^*(x_{\text{coarse}}^{\mathcal{A}}))} \pi_i} = \hat{W}_{\text{coarse}}^{\mathcal{A} \to \mathcal{B}} \in \mathbb{R}^{H_{\text{coarse}} \times W_{\text{coarse}} \times 2}, \qquad (9)$$

where $N_4(k^*)$ denotes the set of k^* and the four closest anchors on the left, right, top, and bottom. We conduct an ablation on the Transformer match decoder in Table 2, and find that it particularly improves results in our full approach, using the loss formulation we propose in Section 3.4.

Decoder Architecture: In early experiments, we found that ConvNet coarse match decoders overfit to the training

---

[2]This ensures that there is no overlap between anchors and no holes in the cover.
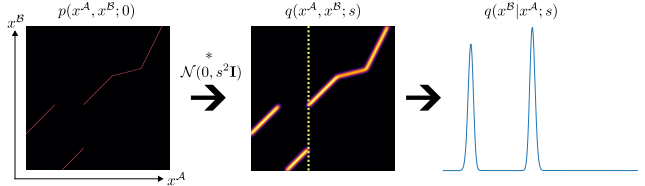

Figure 3. **Illustration of localizability of matches.** At infinite resolution the match distribution can be seen as a 2D surface (illustrated as 1D lines in the figure), however at a coarser scale s this distribution becomes blurred due to motion boundaries. This means it is necessary to both use a model and an objective function capable of representing multimodal distributions.

resolution. Additionally, they tend to be over-reliant on locality. While locality is a powerful cue for refinement, it leads to oversmoothing for the coarse warp. To address this, we propose a transformer decoder without using position encodings. By restricting the model to only propagate by feature similarity, we found that the model became significantly more robust.

The proposed Transformer matcher decoder consists of 5 ViT blocks, with 8 heads, hidden size D 1024, and MLP size 4096. The input is the concatenation of projected DINOv2 [39] features of dimension 512, and the 512-dimensional output of the GP module, which corresponds to the match encoder E_θ proposed in DKM [17]. The output is a vector of $B \times H \times W \times (K + 1)$ where K is the number of classification anchors[3] (parameterizing the conditional distribution $p(x^{\mathcal{B}}|x^{\mathcal{A}})$), and the extra 1 is the matchability score $p^{\mathcal{A}}(x^{\mathcal{A}})$.

3.4. Robust Loss Formulation

Intuition: The conditional match distribution at coarse scales is more likely to exhibit multimodality than during refinement, which is conditional on the previous warp. This means that the coarse matcher needs to model multimodal distributions, which motivates our regression-by-classification approach. In contrast, the refinement of the warp needs only to represent unimodal distributions, which motivates our robust regression loss.

Theoretical Model: We model the matchability at scale s as

$$q(x^{\mathcal{A}}, x^{\mathcal{B}}; s) = \mathcal{N}(0, s^2\mathbf{I}) * p(x^{\mathcal{A}}, x^{\mathcal{B}}; 0). \qquad (10)$$

Here $p(x^{\mathcal{A}}, x^{\mathcal{B}}; 0)$ corresponds to the exact mapping at infinite resolution, and $*$ denotes convolution. This can be interpreted as a diffusion in the localization of the matches over scales. When multiple objects in a scene are projected into images, so-called motion boundaries arise. These are discontinuities in the matches which we illustrate in Figure 3. The diffusion near these motion boundaries causes

---

[3]When used for regression, K is set to $K = 2$, and the decoding to a warp is the identity function.
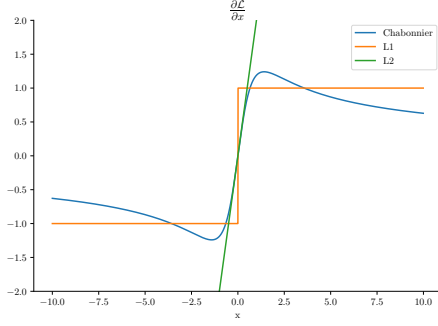
19794

Figure 4. **Comparison of loss gradients.** We use the generalized Charbonnier [3] loss for refinement, which locally matches L2 gradients, but globally decays with $|x|^{-1/2}$ toward zero.

the conditional distribution to become multimodal, explaining the need for multimodality in the coarse global matching. Given an initial choice of $(x^{\mathcal{A}}, x^{\mathcal{B}})$, as in the refinement, the conditional distribution is unimodal locally. However, if the initial choice is far outside the support of the distribution, using a non-robust loss function is problematic. It is therefore motivated to use a robust regression loss for refinement.

Loss formulation: Motivated by intuition and the theoretical model we now propose our loss formulation from a probabilistic perspective, aiming to minimize the Kullback–Leibler divergence between the estimated match distribution at each scale, and the theoretical model distribution at that scale. We begin by formulating the coarse loss. With non-overlapping bins as defined in Section 3.3 the Kullback–Leibler divergence (where terms that are constant w.r.t. θ are ignored) is

$$D_{\mathrm{KL}}(q(x^{\mathcal{B}}, x^{\mathcal{A}}; s) || p_{\mathrm{coarse},\theta}(x^{\mathcal{B}}, x^{\mathcal{A}})) = \quad (11)$$

$$\mathbb{E}_{x^{\mathcal{A}}, x^{\mathcal{B}} \sim q}\left[-\log p_{\mathrm{coarse},\theta}(x^{\mathcal{B}}|x^{\mathcal{A}}) p_{\mathrm{coarse},\theta}(x^{\mathcal{A}}) \right] = \quad (12)$$

$$-\int_{x^{\mathcal{A}}, x^{\mathcal{B}}} \log \pi_{k^{\mathcal{B}}}(x^{\mathcal{A}}) + \log p_{\mathrm{coarse},\theta}(x^{\mathcal{A}}) dq, \quad (13)$$

for $k^{\mathcal{B}} = \mathrm{argmin}_k \|m_k - x^{\mathcal{B}}\|$ the index of the closest anchor to $x^{\mathcal{B}}$. Following DKM [17] we add a hyperparameter λ that controls the weighting of the marginal compared to that of the conditional as

$$-\int_{x^{\mathcal{A}}, x^{\mathcal{B}}} \log \pi_{k^{\mathcal{B}}}(x^{\mathcal{A}}) + \lambda \log p_{\mathrm{coarse},\theta}(x^{\mathcal{A}}) dq. \quad (14)$$

In practice, we approximate q with a discrete set of known correspondences $\mathcal{C}$:

$$\mathcal{L}_{\mathrm{coarse}} = \sum_{x^{\mathcal{A}}, x^{\mathcal{B}} \in \mathcal{C}} -\log \pi_{k^{\mathcal{B}}}(x^{\mathcal{A}}) - \lambda \log p_{\mathrm{coarse},\theta}(x^{\mathcal{A}})$$

$$(15)$$

Furthermore, to be consistent with previous works [17, 56] we use a binary cross-entropy loss on $p_{\mathrm{coarse},\theta}(x^{\mathcal{A}})$. We next discuss the fine loss $\mathcal{L}_{\mathrm{fine}}$.

Table 2. **Ablation study.** We systematically investigate the impact of our contributions, see Section 4.1 for detailed analysis. Measured in 100-percentage correct keypoints (PCK) (lower is better).

Setup $\downarrow$ 100-PCK@ $\rightarrow$	1px	3px	5px
I (Baseline): DKM [17]	17.0	7.3	5.8
II: I, $F_{\mathrm{coarse},\theta} = \mathrm{RN50}$, $F_{\mathrm{fine},\theta} = \mathrm{RN50}$	16.0	6.1	4.5
III: II, $F_{\mathrm{fine},\theta} = \mathrm{VGG19}$	14.5	5.4	4.5
IV: III, $D_\theta = \mathrm{Transformer}$	14.4	5.4	4.1
V: IV, $F_{\mathrm{coarse},\theta} = \mathrm{DINOv2}$	14.3	4.6	3.2
VI: V, $\mathcal{L}_{\mathrm{coarse}} = \mathrm{reg.\text{-}by\text{-}class.}$	13.6	4.1	2.8
VII (Ours): VI, $\mathcal{L}_{\mathrm{refine}} = \mathrm{robust}$	**13.1**	**4.0**	**2.7**
VIII: VII, $D_\theta = \mathrm{ConvNet}$	14.0	4.9	3.5

We model the output of the refinement at scale i as a generalized Charbonnier [3] (with $\alpha = 0.5$) distribution, for which the refiners estimate the mean μ. The generalized Charbonnier distribution behaves locally like a Normal distribution, but has a flatter tail. When used as a loss, the gradients behave locally like L2, but decay towards 0, see Figure 4. Its logarithm, (ignoring terms that do not contribute to the gradient, and up-to-scale) reads

$$\log p_\theta(x_i^{\mathcal{B}}|x_i^{\mathcal{A}}, \hat{W}_{i+1}^{\mathcal{A} \rightarrow \mathcal{B}}) = \quad (16)$$

$$-(||\mu_\theta(x_i^{\mathcal{A}}, \hat{W}_{i+1}^{\mathcal{A} \rightarrow \mathcal{B}}) - x_i^{\mathcal{B}}||^2 + s^2)^{1/4}, \quad (17)$$

where $\mu_\theta(x_i^{\mathcal{A}}, \hat{W}_{i+1}^{\mathcal{A} \rightarrow \mathcal{B}})$ is the estimated mean of the distribution, and $s = 2^i c$. In practice, we choose $c = 0.03$. The Kullback–Leibler divergence for each fine scale $i \in \{0, 1, 2, 3\}$ (where terms that are constant with respect to θ are ignored) reads

$$D_{\mathrm{KL}}(q(x_i^{\mathcal{B}}, x_i^{\mathcal{A}}; s = 2^i c) || p_{i,\theta}(x_i^{\mathcal{B}}, x_i^{\mathcal{A}}|\hat{W}_{i+1}^{\mathcal{A} \rightarrow \mathcal{B}})) = \quad (18)$$

$$\mathbb{E}_{x_i^{\mathcal{A}}, x_i^{\mathcal{B}} \sim q}\left[(||\mu_\theta(x_i^{\mathcal{A}}, \hat{W}_{i+1}^{\mathcal{A} \rightarrow \mathcal{B}}) - x_i^{\mathcal{B}}||^2 + s^2)^{1/4} \right] +$$

$$\mathbb{E}_{x_i^{\mathcal{A}}, x_i^{\mathcal{B}} \sim q}\left[-\log p_{i,\theta}(x_i^{\mathcal{A}}|\hat{W}_{i+1}^{\mathcal{A} \rightarrow \mathcal{B}}) \right]. \quad (19)$$

In practice, we approximate q with a discrete set of known correspondences $\mathcal{C}$,

$$\mathcal{L}_{\mathrm{fine}} = \sum_i \sum_{x_i^{\mathcal{A}}, x_i^{\mathcal{B}} \in \mathcal{C}} (||\mu_\theta(x_i^{\mathcal{A}}, \hat{W}_{i+1}^{\mathcal{A} \rightarrow \mathcal{B}}) - x_i^{\mathcal{B}}||^2 + s^2)^{1/4}$$

$$-\lambda \log p_{i,\theta}(x_i^{\mathcal{A}}|\hat{W}_{i+1}^{\mathcal{A} \rightarrow \mathcal{B}})$$

$$(20)$$

and use a binary cross-entropy loss on $p_{i,\theta}(x_i^{\mathcal{A}}|\hat{W}_{i+1}^{\mathcal{A} \rightarrow \mathcal{B}})$, we additionally utilize clipping as in DKM [17], further details are found in the supplementary.

Our combined loss yields:

$$\mathcal{L} = \mathcal{L}_{\mathrm{coarse}} + \mathcal{L}_{\mathrm{fine}}. \quad (21)$$

Table 3. **SotA comparison on IMC2022 [26].** Measured in mAA (higher is better).

Method ↓	mAA →	@10 ↑
SiLK [22]		68.6
SP [15]+SuperGlue [44]		72.4
LoFTR [48] CVPR'21		78.3
MatchFormer [59] ACCV'22		78.3
QuadTree [50] ICLR'22		81.7
ASpanFormer [13] ECCV'22		83.8
DKM [17] CVPR'23		83.1
RoMa		**88.0**

Table 4. **SotA comparison on WxBS [37].** Measured in mAA at 10px (higher is better).

Method	mAA@ →	10px ↑
DISK [57] NeurIps'20		35.5
DISK + LightGlue [33, 57] ICCV'23		41.7
SuperPoint +SuperGlue [15, 44] CVPR'20		31.4
LoFTR [48] CVPR'21		55.4
DKM [17] CVPR'23		58.9
RoMa		**80.1**

4. Experiments

4.1. Ablation Study

Here we investigate the impact of our contributions. We conduct all our ablations on a validation test detailed in the supplementary.

Setup I consists of the same components as in DKM [17], retrained by us. In Setup II we do not share weights between the fine and coarse features, which improves performance due to specialization of the features. In Setup III we replace the RN50 fine features with a VGG19, which further improves performance. This is intriguing, as VGG19 features are worse performing when used as coarse features as we show in Table 1. We then add the proposed Transformer match decoder in Setup IV, however using the baseline regression approach. Further, we incorporate the DINOv2 coarse features in Setup V, this gives a significant improvement, owing to their significant robustness. Next, in Setup VI change the loss function and output representation of the Transformer match decoder D_θ to regression-by-classification, and next in Setup VII use the robust regression loss. Both these changes further significantly improve performance. This setup constitutes RoMa. When we change back to the original ConvNet match decoder in Setup VIII from this final setup, we find that the performance significantly drops, showing the importance of the proposed Transformer match decoder.

Table 5. **SotA comparison on MegaDepth-1500 [30, 48].** Measured in AUC (higher is better).

Method ↓	AUC@ →	5° ↑	10° ↑	20° ↑
LightGlue [33] ICCV'23		51.0	68.1	80.7
LoFTR [48] CVPR'21		52.8	69.2	81.2
PDC-Net+ [56] TPAMI'23		51.5	67.2	78.5
ASpanFormer [13] ECCV'22		55.3	71.5	83.1
ASTR [65] CVPR'23		58.4	73.1	83.8
DKM [17] CVPR'23		60.4	74.9	85.1
PMatch [67] CVPR'23		61.4	75.7	85.7
CasMTR [11] ICCV'23		59.1	74.3	84.8
RoMa		**62.6**	**76.7**	**86.3**

Table 6. **SotA comparison on ScanNet-1500 [14, 44].** Measured in AUC (higher is better).

Method ↓	AUC@ →	5° ↑	10° ↑	20° ↑
SuperGlue [44] CVPR'19		16.2	33.8	51.8
LoFTR [48] CVPR'21		22.1	40.8	57.6
PDC-Net+ [56] TPAMI'23		20.3	39.4	57.1
ASpanFormer [13] ECCV'22		25.6	46.0	63.3
PATS [38] CVPR'23		26.0	46.9	64.3
DKM [17] CVPR'23		29.4	50.7	68.3
PMatch [67] CVPR'23		29.4	50.1	67.4
CasMTR [11] ICCV'23		27.1	47.0	64.4
RoMa		**31.8**	**53.4**	**70.9**

4.2. Training Setup

We use the training setup as in DKM [17]. Following DKM, we use a canonical learning rate (for batch-size 8) of 10^{-4} for the decoder, and $5 \cdot 10^{-6}$ for the encoder(s), which we scale linearly with batch-size. We use the same training split as in DKM, which consists of randomly sampled pairs from the MegaDepth and ScanNet sets excluding the scenes used for testing. The supervised warps are derived from dense depth maps from multi-view-stereo (MVS) of SfM reconstructions in the case of MegaDepth, and from RGB-D for ScanNet. Following previous work [13, 17, 48], we use a model trained on the ScanNet training set when evaluating on ScanNet-1500. All other evaluation is done on a model trained only on MegaDepth. As in DKM we train both the coarse matching and refinement networks jointly. Note that since we detach gradients between the coarse matching and refinement, the network could in principle also be trained in two stages. For results used in the ablation, we used a resolution of 448×448, and for the final method we trained on a resolution of 560×560.

4.3. Two-View Geometry

We evaluate on a diverse set of two-view geometry benchmarks. We follow DKM [17] and sample correspondences

Table 7. **SotA comparison on Megadepth-8-Scenes [17].** Measured in AUC (higher is better).

Method ↓ AUC →	@5°	@10°	@20°
PDCNet+ [56] TPAMI'23	51.8	66.6	77.2
ASpanFormer [13] ECCV'22	57.2	72.1	82.9
DKM [17] CVPR'23	60.5	74.5	84.2
RoMa	**62.2**	**75.9**	**85.3**

Table 8. **SotA comparison on InLoc [49].** We report the percentage of query images localized within 0.25/0.5/1.0 meters and 2/5/10° of the ground-truth pose (higher is better).

Method	DUC1	DUC2
	(0.25m,2°)/(0.5m,5°)/(1.0m,10°)	
PATS	55.6 / 71.2 / 81.0	58.8 / 80.9 / 85.5
DKM	51.5 / 75.3 / 86.9	63.4 / 82.4 / **87.8**
CasMTR	53.5 / 76.8 / 85.4	51.9 / 70.2 / 83.2
RoMa	**60.6 / 79.3 / 89.9**	**66.4 / 83.2 / 87.8**

using a balanced sampling approach, producing 10,000 matches for estimation. We consistently improve compared to prior work across the board, in particular achieving a relative error reduction on the competitive IMC2022 [26] benchmark by 26%, and a gain of 36% in performance on the exceptionally difficult WxBS [37] benchmark.

Image Matching Challenge 2022: We submit to the 2022 version of the image matching challenge [26], which consists of a hidden test-set of Google street-view images with the task to estimate the fundamental matrix between them. We present results in Table 3. RoMa attains significant improvements compared to previous approaches, with a relative error reduction of 26% compared to the previous best approach.

WxBS Benchmark: We evaluate RoMa on the extremely difficult WxBS benchmark [37], version 1.1 with updated ground truth and evaluation protocol[4]. The metric is mean average precision on ground truth correspondences consistent with the estimated fundamental matrix at a 10 pixel threshold. All methods use MAGSAC++ [2] as implemented in OpenCV. Results are presented in Table 4. Here we achieve an outstanding improvement of 36% compared to the state-of-the-art. We attribute these major gains to the superior robustness of RoMa compared to previous approaches. We qualitatively present examples of this in the supplementary.

MegaDepth-1500 Pose Estimation: We use the MegaDepth-1500 test set [48] which consists of 1500 pairs from scene 0015 (St. Peter's Basilica) and 0022 (Brandenburger Tor). We follow the protocol in [13, 48] and use a RANSAC threshold of 0.5. Results are presented in Table 5.

ScanNet-1500 Pose Estimation: ScanNet [14] is a large scale indoor dataset, composed of challenging sequences with low texture regions and large changes in perspective. We follow the evaluation in SuperGlue [44]. Results are presented in Table 6. We achieve state-of-the-art results, achieving the first AUC@20° scores over 70.

MegaDepth-8-Scenes: We evaluate RoMa on the Megadepth-8-Scenes benchmark [17, 30]. We present results in Table 7. Here too we outperform previous approaches.

4.4. Visual Localization

We evaluate RoMa on the InLoc [49] Visual Localization benchmark, using the HLoc [43] pipeline. We follow the approach in DKM [17] to sample correspondences. Results are presented in Table 8. We show large improvements compared to all previous approaches, setting a new state-of-the-art. We show results on Aachen v1.1 in the supplementary.

5. Conclusion

We have presented RoMa, a robust dense feature matcher. Our model leverages frozen pretrained coarse features from the foundation model DINOv2 together with specialized ConvNet fine features, creating a precisely localizable and robust feature pyramid. We further improved performance with our proposed tailored transformer match decoder, which predicts anchor probabilities instead of regressing coordinates. Finally, we proposed an improved loss formulation through regression-by-classification with subsequent robust regression. Our comprehensive experiments show that RoMa achieves major gains across the board, setting a new state-of-the-art. In particular, our biggest gains (36% increase on WxBS [37]) are achieved on the most difficult benchmarks, highlighting the robustness of our approach. Code is provided at github.com/Parskatt/RoMa.

---

[4]https://ducha-aiki.github.io/wide-baseline-stereo-blog/2021/07/30/Reviving-WxBS-benchmark

References

[1] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. HPatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5173–5182, 2017.

[2] Daniel Barath, Jana Noskova, Maksym Ivashechkin, and Jiri Matas. MAGSAC++, a fast, reliable and accurate robust estimator. In *Conference on Computer Vision and Pattern Recognition*, 2020. 8

[3] Jonathan T Barron. A general and adaptive robust loss function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4331–4339, 2019. 3, 6

[4] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006. 3

[5] Michael J Black and Paul Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer vision and image understanding*, 63(1):75–104, 1996. 3

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

[7] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021. 3

[8] Ignas Budvytis, Marvin Teichmann, Tomas Vojir, and Roberto Cipolla. Large scale joint semantic re-localisation and scene understanding via globally unique instance coordinate regression. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 86.1–86.13. BMVA Press, 2019. 3

[9] Georg Bökman and Fredrik Kahl. A case for using rotation invariant features in state of the art feature matchers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5110–5119, 2022. 3

[10] Georg Bökman, Johan Edstedt, Michael Felsberg, and Fredrik Kahl. Steerers: A framework for rotation equivariant keypoint descriptors. In *CVPR*, 2024. 3

[11] Chenjie Cao and Yanwei Fu. Improving transformer-based image matching by cascaded capturing spatially informative keypoints. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12129–12139, 2023. 7

[12] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 3

[13] Hongkai Chen, Zixin Luo, Lei Zhou, Yurun Tian, Mingmin Zhen, Tian Fang, David Mckinnon, Yanghai Tsin, and Long Quan. ASpanFormer: Detector-free image matching with adaptive span transformer. In *Proc. European Conference on Computer Vision (ECCV)*, 2022. 3, 7, 8

[14] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 7, 8

[15] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018. 3, 7

[16] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-Net: A Trainable CNN for Joint Detection and Description of Local Features. In *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 4

[17] Johan Edstedt, Ioannis Athanasiadis, Mårten Wadenbäck, and Michael Felsberg. DKM: Dense kernelized feature matching for geometry estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 1, 2, 3, 4, 5, 6, 7, 8

[18] Johan Edstedt, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. DeDoDe: Detect, Don't Describe – Describe, Don't Detect for Local Feature Matching. In *2024 International Conference on 3D Vision (3DV)*. IEEE, 2024. 3

[19] Michael Felsberg, P-E Forssen, and H Scharr. Channel smoothing: Efficient robust smoothing of low-level signal features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(2):209–222, 2006. 3

[20] Divyansh Garg, Yan Wang, Bharath Hariharan, Mark Campbell, Kilian Q Weinberger, and Wei-Lun Chao. Wasserstein distances for stereo disparity estimation. *Advances in Neural Information Processing Systems*, 33:22517–22529, 2020. 3

[21] Hugo Germain, Vincent Lepetit, and Guillaume Bourmaud. Neural reprojection error: Merging feature learning and camera pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 414–423, 2021. 3

[22] Pierre Gleize, Weiyao Wang, and Matt Feiszli. SiLK: Simple Learned Keypoints. In *ICCV*, 2023. 7

[23] Gustav Häger, Mikael Persson, and Michael Felsberg. Predicting disparity distributions. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4363–4369. IEEE, 2021. 3

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4

[25] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16000–16009, 2022. 1

[26] Addison Howard, Eduard Trulls, Kwang Moo Yi, Dmitry Mishkin, Sohier Dane, and Yuhe Jin. Image matching challenge 2022, 2022. 7, 8

[27] Jan J Koenderink. The structure of images. *Biological cybernetics*, 50(5):363–370, 1984. 2

[28] Xinghui Li, Kai Han, Shuda Li, and Victor Prisacariu. Dual-resolution correspondence networks. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020. 3

[29] Xiaotian Li, Shuzhe Wang, Yi Zhao, Jakob Verbeek, and Juho Kannala. Hierarchical scene coordinate classification and regression for visual localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11983–11992, 2020. 3

[30] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2041–2050, 2018. 4, 7, 8

[31] Yutong Lin, Ze Liu, Zheng Zhang, Han Hu, Nanning Zheng, Stephen Lin, and Yue Cao. Could giant pre-trained image models extract universal representations? *Advances in Neural Information Processing Systems*, 35:8332–8346, 2022. 1

[32] Tony Lindeberg. Scale-space theory: A basic tool for analyzing structures at different scales. *Journal of applied statistics*, 21(1-2):225–270, 1994. 2

[33] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local Feature Matching at Light Speed. In *ICCV*, 2023. 3, 7

[34] Haisong Liu, Tao Lu, Yihui Xu, Jia Liu, Wenjie Li, and Lijun Chen. Camliflow: bidirectional camera-lidar fusion for joint optical flow and scene flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5791–5801, 2022. 3

[35] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 3

[36] Iaroslav Melekhov, Aleksei Tiulpin, Torsten Sattler, Marc Pollefeys, Esa Rahtu, and Juho Kannala. Dgc-net: Dense geometric correspondence network. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1034–1042. IEEE, 2019. 3

[37] Dmytro Mishkin, Jiri Matas, Michal Perdoch, and Karel Lenc. WxBS: Wide Baseline Stereo Generalizations. In *Proceedings of the British Machine Vision Conference*. BMVA, 2015. 1, 7, 8

[38] Junjie Ni, Yijin Li, Zhaoyang Huang, Hongsheng Li, Hujun Bao, Zhaopeng Cui, and Guofeng Zhang. Pats: Patch area transportation with subdivision for local feature matching. In *The IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2023. 1, 3, 7

[39] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *arXiv:2304.07193*, 2023. 1, 2, 3, 5

[40] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005. 4, 1

[41] Jerome Revaud, Cesar De Souza, Martin Humenberger, and Philippe Weinzaepfel. R2d2: Reliable and repeatable detector and descriptor. *Advances in neural information processing systems*, 32:12405–12415, 2019. 3

[42] I. Rocco, M. Cimpoi, R. Arandjelović, A. Torii, T. Pajdla, and J. Sivic. Neighbourhood consensus networks. In *Proceedings of the 32nd Conference on Neural Information Processing Systems*, 2018. 3

[43] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12716–12725, 2019. 1, 8, 2, 4

[44] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020. 1, 3, 7, 8

[45] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3247–3257, 2021. 4

[46] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, et al. Benchmarking 6dof outdoor visual localization in changing conditions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8601–8610, 2018. 2

[47] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 1

[48] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8922–8931, 2021. 1, 3, 7, 8

[49] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. Inloc: Indoor visual localization with dense matching and view synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7199–7209, 2018. 8

[50] Shitao Tang, Jiahui Zhang, Siyu Zhu, and Ping Tan. Quadtree attention for vision transformers. In *International Conference on Learning Representations*, 2022. 3, 7

[51] Zhuotao Tian, Hengshuang Zhao, Michelle Shu, Zhicheng Yang, Ruiyu Li, and Jiaya Jia. Prior guided feature enrichment network for few-shot segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 44(2):1050–1065, 2020. 1

[52] Luís Torgo and João Gama. Regression by classification. In *Advances in Artificial Intelligence*, pages 51–60, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg. 3

[53] Prune Truong, Martin Danelljan, Luc V Gool, and Radu Timofte. GOCor: Bringing Globally Optimized Correspondence Volumes into Your Neural Network. *Advances in Neural Information Processing Systems*, 33, 2020. 1

[54] Prune Truong, Martin Danelljan, and Radu Timofte. GLU-Net: Global-local universal network for dense flow and correspondences. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6258–6268, 2020. 3

[55] Prune Truong, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning accurate dense correspondences and when to trust them. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5714–5724, 2021. 3

[56] Prune Truong, Martin Danelljan, Radu Timofte, and Luc Van Gool. PDC-Net+: Enhanced Probabilistic Dense Correspondence Network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 1, 4, 6, 7, 8

[57] Michal J. Tyszkiewicz, Pascal Fua, and Eduard Trulls. DISK: learning local features with policy gradient. In *NeurIPS*, 2020. 3, 7

[58] Cristina Vasconcelos, Vighnesh Birodkar, and Vincent Dumoulin. Proper reuse of image classification features improves object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13628–13637, 2022. 1

[59] Qing Wang, Jiaming Zhang, Kailun Yang, Kunyu Peng, and Rainer Stiefelhagen. MatchFormer: Interleaving attention in transformers for feature matching. In *Asian Conference on Computer Vision*, 2022. 7

[60] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14668–14678, 2022. 1

[61] Sholom M. Weiss and Nitin Indurkhya. Rule-based regression. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence. Chambéry, France, August 28 - September 3, 1993*, pages 1072–1078. Morgan Kaufmann, 1993. 3

[62] Sholom M. Weiss and Nitin Indurkhya. Rule-based machine learning methods for functional prediction. *J. Artif. Intell. Res.*, 3:383–403, 1995. 3

[63] Andrew P. Witkin. Scale space filtering. *Proc. 8th International Joint on Artificial Intelligence*, pages 1091–1022, 1983. 2

[64] Zhenda Xie, Zigang Geng, Jingcheng Hu, Zheng Zhang, Han Hu, and Yue Cao. Revealing the dark secrets of masked image modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14475–14485, 2023. 2

[65] Jiahuan Yu, Jiahao Chang, Jianfeng He, Tianzhu Zhang, Jiyang Yu, and Wu Feng. ASTR: Adaptive spot-guided transformer for consistent local feature matching. In *The IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2023. 7

[66] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. In *International Conference on Learning Representations*, 2022. 1, 3

[67] Shengjie Zhu and Xiaoming Liu. PMatch: Paired masked image modeling for dense geometric matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 3, 7