HANDLING MULTIPLE HYPOTHESES IN COARSE-TO-FINE DENSE IMAGE MATCHING

[1]*Matthieu Vilain,* [1]*Remi Giraud,* [1]*Yannick Berthoumieu, and* [1]*Guillaume Bourmaud*

[1]Univ. Bordeaux, CNRS, Bordeaux INP, IMS, UMR 5218, F-33400 Talence, France

ABSTRACT

Dense image matching aims to find a correspondent for every pixel of a source image in a partially overlapping target image. State-of-the-art methods typically rely on a coarse-to-fine mechanism where a single correspondent hypothesis is produced per source location at each scale. In challenging cases – such as at depth discontinuities or when the target image is a strong zoom-in of the source image – the correspondents of neighboring source locations are often widely spread and predicting a single correspondent hypothesis per source location at each scale may lead to erroneous matches. In this paper, we investigate the idea of predicting *multiple* correspondent hypotheses per source location at each scale instead. We consider a beam search strategy to propagate multiple hypotheses at each scale and propose integrating these multiple hypotheses into cross-attention layers, resulting in a novel dense matching architecture called BEAMER. BEAMER learns to preserve and propagate multiple hypotheses across scales, making it significantly more robust than state-of-the-art methods, especially at depth discontinuities or when the target image is a strong zoom-in of the source image. Our code will be made publicly available.

***Index Terms*—** Image Matching, Transformer, Multiple Hypotheses, Dense Matching

1. INTRODUCTION

Image matching seeks to establish correspondences between a pair of partially overlapping source and target images. This task is fundamental to various computer vision applications, including 3D reconstruction [1], simultaneous localization and mapping [2], and visual localization [3].

Image matching methods usually fall into one of the following three categories. *Detector-based* methods [4, 5, 6, 7, 8, 9] establish correspondences between keypoints detected in the source image and in the target image. *Semi-dense methods* [10, **?**, 11, 12, 13], establish correspondences between source keypoints located on a coarse regular grid (hence the name "semi-dense") and the whole target image. *Dense methods* [14, 15, 16, 17, 18, 19], try to find a correspondent for every pixel of the source image in the target image. Dense methods currently outperform both detector-based methods and semi-dense methods on pose estimation benchmarks [3, 20, 21, 22]. Therefore, in the rest of the paper, we focus on dense methods.

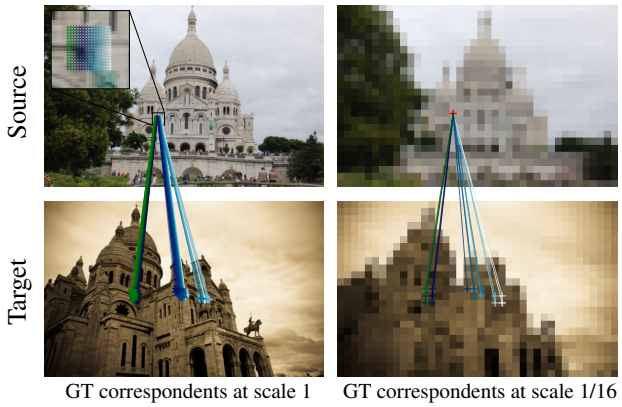

Fig. 1: **(Left)** Source locations in the neighborhood of a depth discontinuity (top-left: 16x16 patch) have ground truth (GT) correspondents in the target image that are widely spread. Here, the GT correspondents (bottom-left) are spread across three different modes in the target image. **(Right)** At scale 1/16, the source location + (top-right), that corresponds to the 16x16 patch (top-left) at scale 1, has 15 GT correspondents that are widely spread (bottom-right). Thus, state-of-the-art dense image matching methods that rely on a coarse-to-fine mechanism and predict a single correspondent hypothesis per source location at each scale, have difficulties correctly establishing correspondences at depth discontinuities.

In this dense setting, most methods rely on a *coarse-to-fine* mechanism to efficiently search for the correspondent of each source pixel in the target image. In practice, at each scale of the coarse-to-fine mechanism, a *single* correspondent hypothesis is produced per source location. This approach works well when there is no ambiguity, *i.e.* when neighboring source locations at a given scale have target correspondents all within a small neighborhood. However, in challenging cases – such as at depth discontinuities or when the target image is a strong zoom-in of the source image – the correspondents of neighboring source locations are often widely spread (see Fig. 1) and predicting a single correspondent hypothesis per source location at each scale may lead to erroneous matches, as acknowledged in the limitations of [16]. In this paper, we investigate the idea of predicting *multiple correspondent hypotheses* per source location at each scale.

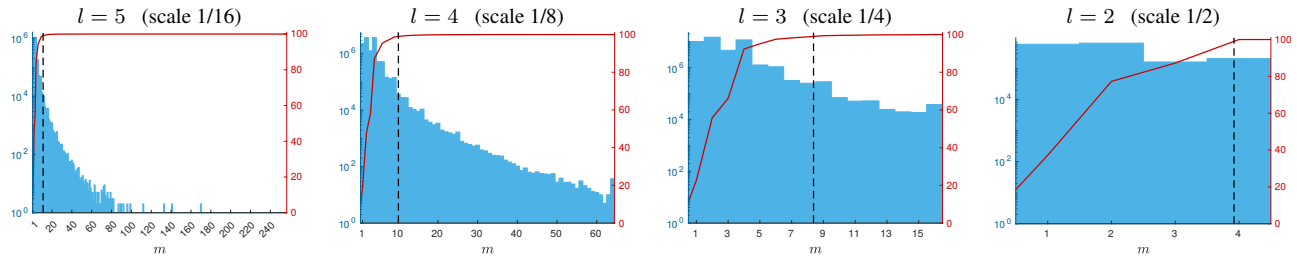

Fig. 2: **Experimental multiple hypothesis analysis on MegaDepth training samples.** For each coarse-to-fine scale, the histogram indicates the number of times a one-to-m correspondence problem was found in the ground truth correspondences (left axis in log-scale). The red curve (right axis) is the cumulative histogram. The vertical dashed line marks the value of m that encompasses 99% of the distribution.

Our contributions are as follows - (i) We formulate dense matching as a series of coarse-to-fine classification steps, which leads us to consider a beam search strategy [23] for propagating multiple hypotheses at each scale. (ii) We propose integrating these multiple hypotheses into cross-attention layers, resulting in a novel dense matching architecture, called BEAMER, which is able to learn to preserve and propagate multiple hypotheses across scales. (iii) Our experiments show that the performance of state-of-the-art dense matching methods significantly decreases when the correspondents of neighboring source locations are widely spread. BEAMER is much more robust in these cases and significantly outperforms dense matching methods.

2. METHOD

2.1. A series of coarse-to-fine classification steps

Dense image matching (DIM) aims to establish correspondences between a source image $\mathtt{I}_\mathrm{s}$ ($H_\mathrm{s} \times W_\mathrm{s} \times 3$) and a target image $\mathtt{I}_\mathrm{t}$ ($H_\mathrm{t} \times W_\mathrm{t} \times 3$). Given a neural network, we extract multi-scale dense feature maps for the source and target images, denoted as $\{\mathtt{F}^l_\mathrm{s}\}_{l=1\ldots L}$ and $\{\mathtt{F}^l_\mathrm{t}\}_{l=1\ldots L}$. At scale l, $\mathtt{F}^l_\mathrm{s}$ and $\mathtt{F}^l_\mathrm{t}$ are of size $\frac{H_\mathrm{s}}{2^{l-1}} \times \frac{W_\mathrm{s}}{2^{l-1}} \times C_l$ and $\frac{H_\mathrm{t}}{2^{l-1}} \times \frac{W_\mathrm{t}}{2^{l-1}} \times C_l$, respectively. The corresponding grids of feature locations at scale l, Ω^l_s and Ω^l_t, are of size $\frac{H_\mathrm{s}}{2^{l-1}} \times \frac{W_\mathrm{s}}{2^{l-1}}$ and $\frac{H_\mathrm{t}}{2^{l-1}} \times \frac{W_\mathrm{t}}{2^{l-1}}$. The objective of DIM is to compute correspondences between fine-scale source locations $\{\mathbf{p}^1_{\mathrm{s},i}\}_{i=1,\ldots,|\Omega^1_\mathrm{s}|}$ and fine-scale target locations $\{\mathbf{p}^1_{\mathrm{t},i}\}_{i=1,\ldots,|\Omega^1_\mathrm{t}|}$, where $\mathbf{p}$ is a 2D vector of integers.

To address the computational challenges of DIM, the task can be formulated as a series of coarse-to-fine classification steps. At the coarsest scale L, dense correspondence maps $\mathtt{C}^L_{\mathbf{p}^L_{\mathrm{s},i}} = \mathrm{softmax}(\mathtt{F}^L_\mathrm{s}(\mathbf{p}^L_{\mathrm{s},i}) \odot \mathtt{F}^L_\mathrm{t})$ are computed over the entire target grid Ω^L_t for each source location $\mathbf{p}^L_{\mathrm{s},i} = \left\lceil \frac{\mathbf{p}^1_{\mathrm{s},i}}{2^{L-1}} \right\rceil \in \Omega^L_\mathrm{s}$.

The process then iterates to finer scales. For each source location $\mathbf{p}^l_{\mathrm{s},i} = \left\lceil \frac{\mathbf{p}^1_{\mathrm{s},i}}{2^{l-1}} \right\rceil \in \Omega^l_\mathrm{s}$, the coarser correspondence map $\mathtt{C}^{l+1}_{\mathbf{p}^{l+1}_{\mathrm{s},i}}$ is used to determine a sparse search region $\Omega^l_{\mathrm{t},i} \subset \Omega^l_\mathrm{t}$ (see Sec. 2.2). Within this region, a sparse correspondence map is computed as:

$$\mathtt{C}^l_{\mathbf{p}^l_{\mathrm{s},i}} = \mathrm{softmax}(\mathtt{F}^l_\mathrm{s}(\mathbf{p}^l_{\mathrm{s},i}) \odot \mathtt{F}^l_\mathrm{t}(\Omega^l_{\mathrm{t},i})). \tag{1}$$

This refinement continues down to the finest scale (l=1). The final sparse correspondence map $\mathtt{C}^1_{\mathbf{p}^1_{\mathrm{s},i}}$ is computed, and the estimated correspondent $\mathbf{p}^1_{\mathrm{t},i}$ of $\mathbf{p}^1_{\mathrm{s},i}$ is defined as the expectation of $\mathtt{C}^1_{\mathbf{p}^1_{\mathrm{s},i}}$.

2.2. Beam Search

A key step in the previously described framework is the definition of the sparse search region $\Omega^l_{\mathrm{t},i}$ at each scale. A simple strategy is to take the argmax (top 1) of $\mathtt{C}^{l+1}_{\mathbf{p}^{l+1}_{\mathrm{s},i}}$ with a small local window around it [24, 12], but this strategy would precisely fail when the correspondents are widely spread (see Fig. 1). Instead, we consider the top K_{l+1} locations of $\mathtt{C}^{l+1}_{\mathbf{p}^{l+1}_{\mathrm{s},i}}$ as the set of 2D locations $\Omega^l_{\mathrm{t},i}$. Such a strategy is called a beam search [23]. More precisely, each location $\mathbf{p}$ from the top K_{l+1} locations is transformed into four locations: $(2\mathbf{p}+[0\,0]^\top, 2\mathbf{p}+[1\,0]^\top, 2\mathbf{p}+[0\,1]^\top, 2\mathbf{p}+[1\,1]^\top)$. Thus, in practice, a sparse correspondence map at scale l is evaluated at $4K_{l+1}$ locations.

Compared to state-of-the-art dense matching methods that predict a single correspondent hypothesis per source location at each scale, our beam search strategy ensures that multiple hypotheses are preserved and propagated across scales, addressing ambiguities effectively.

To make sure this beam search strategy is computationally feasible, we conducted a multiple hypothesis analysis (see Fig. 2). We took 10,000 training image pairs from the MegaDepth dataset [25] with Ground Truth (GT) correspondences $\left\{\left(\mathbf{p}^{\mathrm{GT},1}_{\mathrm{s},k}, \mathbf{p}^{\mathrm{GT},1}_{\mathrm{t},k}\right)\right\}$ computed using the available depth maps and camera poses. For each scale $l = 2, \ldots, 5$, the GT correspondences are down-sampled as:

$$\mathbf{p}^{\mathrm{GT},l}_{\mathrm{s},k} = \left\lceil \frac{\mathbf{p}^{\mathrm{GT},1}_{\mathrm{s},k}}{2^{l-1}} \right\rceil, \quad \mathbf{p}^{\mathrm{GT},l}_{\mathrm{t},k} = \left\lceil \frac{\mathbf{p}^{\mathrm{GT},1}_{\mathrm{t},k}}{2^{l-1}} \right\rceil. \tag{2}$$

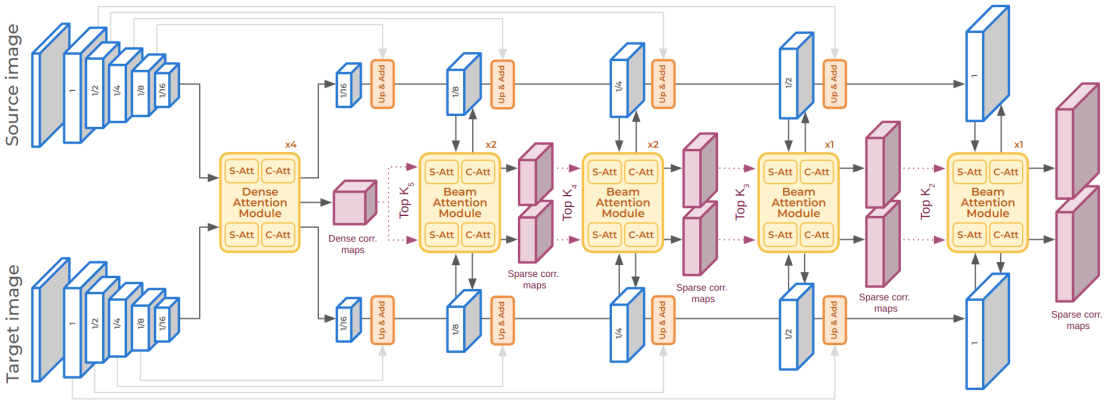

Fig. 3: **BEAMER architecture.** Our novel architecture establishes dense correspondences bidirectionally, in a coarse-to-fine manner. At the coarsest scale, dense correspondence maps are computed that allow to initialize the beam search. Then at each scale, the beam-attention module employs the most promising correspondents from the previous correspondence maps in sparse cross-attention layers that let the features communicate with each other, and produce sparse correspondence maps. Sparse self-attention layers are also implemented. More details are available in the supplementary material.

We experimentally observe that it is possible to properly establish 99% of the correspondences with $K_2 = 4$, $K_3 = 9$, $K_4 = 10$ and $K_5 = 12$, which is indicates that the beam search can be a very efficient strategy. In practice, as we are not provided with *perfect* features, higher values are required. Through experiments, we found $K_2 = 8, K_3 = 16, K_4 = 24, K_5 = 32$ to be a good trade-off between memory requirements and the capacity to correctly establish correspondences (see Ablation study Fig. 6).

2.3. Network Architecture

Our BEAMER architecture is built on a Siamese Feature Pyramid Network with a ResNet-18 backbone. This backbone extracts multi-scale dense feature maps for the source and target images, denoted as $\{\mathrm{F}_{\mathrm{s}}^{l}\}_{l=1\ldots5}$ and $\{\mathrm{F}_{\mathrm{t}}^{l}\}_{l=1\ldots5}$, which serve as the input for the beam search mechanism.

Attention Mechanisms - At the coarsest scale ($l = 5$), BEAMER employs a dense attention module composed of vanilla self- and cross-attention layers. These layers enable the coarse-scale source and target features to communicate and adapt to each other, laying the foundation for accurate correspondence estimation. However, applying dense attention at finer scales is computationally prohibitive due to the resolution of the feature maps.

To address this problem, at each refinement scale ($l < 5$), for each source location $\mathbf{p}_{\mathrm{s},i}^{l}$, BEAMER leverages the sparse search region $\Omega_{\mathrm{t},i}^{l}$ (*i.e.* the top K_{l+1} locations identified by the beam search at scale $l+1$) to perform *sparse cross-attention*. For each feature $\mathrm{F}_{\mathrm{s}}^{l}(\mathbf{p}_{\mathrm{s},i}^{l})$ of the source, sparse cross-attention is computed over its corresponding search space $\mathrm{F}_{\mathrm{t}}^{l}(\Omega_{\mathrm{t},i}^{l})$ as determined by the beam search, allowing the network to effectively mitigate ambiguities caused by multiple hypotheses. In addition, BEAMER incorporates sparse self-attention layers, which applies the same principle of beam search within the source features and within the target features independently. This process, independent of the correspondence search, enables efficient self-attention while preserving computational feasibility. Together, these operations form a *beam attention module*, which consists of two sparse self-attention layers and two sparse cross-attention layers, efficiently capturing both local and global information at every scale. Technical details are provided in supplementary material.

Overall Architecture - The BEAMER architecture (see Fig. 3) combines dense and beam attention modules to efficiently preserve and propagate multiple hypotheses across scales. At the coarsest scale ($l = 5$), 4 dense attention modules are applied. For the refinement scales ($l = 4$ to $l = 1$), BEAMER uses $[2, 2, 1, 1]$ beam attention modules, respectively. At each refinement scale, sparse correspondence maps are computed after the beam attention modules. All operations are performed bidirectionally, *i.e.* source $\rightarrow$ target and target $\rightarrow$ source, to predict correspondents for all pixels in both the source and target images. This design ensures that BEAMER leverages the strengths of dense attention at the coarse scale while efficiently resolving ambiguities at finer scales through beam attention.

2.4. Training

At training-time, we are provided with image pairs and GT correspondences. For each source/target image pair, a set of GT correspondences $\left\{(\mathbf{p}_{\mathrm{s},k}^{\mathrm{GT},1}, \mathbf{p}_{\mathrm{t},k}^{\mathrm{GT},1})\right\}_{k=1\ldots N}$ is available. Our objective is to maximize the likelihood of each correspondence at each scale $l = 1\ldots L$ to learn to preserve and propagate multiple hypotheses across scales. In our classification framework, this is equivalent to minimizing a sum of negative log-likelihood (*a.k.a.* cross-entropy) terms:

Table 1: **Matching accuracy at 3 pixels on MegaDepth-1500 [5], MegaDepth-8-scenes [16] and HPatches [21] for increasing *spread* levels (η in pixels).** BEAMER consistently outperforms the state-of-the-art methods: the more the spreading increases, the greater the gap between BEAMER and state-of-the-art methods grows. The gap is smaller on HPatches as the image pairs are less challenging (planar scenes).

Method	Matching Accuracy @3pix (%) ↑											
	MegaDepth-8scenes				**HPatches**				**MegaDepth-1500**			
	η∈[20,40]	[40,60]	[60,80]	[80,100]	η∈[20,40]	[40,60]	[60,80]	[80,100]	η∈[20,40]	[40,60]	[60,80]	[80,100]
EcoTR [24] ECCV'22	90.4	68.6	53.1	43.1	63.0	51.1	28.3	12.4	85.9	68.3	54.8	48.3
CasMTR [12] ICCV'23	92.4	71.0	57.8	47.4	69.9	57.6	49.5	35.2	92.5	72.7	56.8	51.0
DKM [16] CVPR'23	93.0	71.7	54.9	45.4	70.9	58.4	50.4	37.4	93.4	75.3	60.2	53.9
RoMa [19] CVPR'24	**95.3**	77.5	60.8	51.2	72.8	61.3	56.8	40.1	94.5	78.3	64.8	59.3
BEAMER	94.9	**83.5**	**77.2**	**69.2**	**73.9**	**63.5**	**59.7**	**45.2**	**94.7**	**83.3**	**77.6**	**72.2**

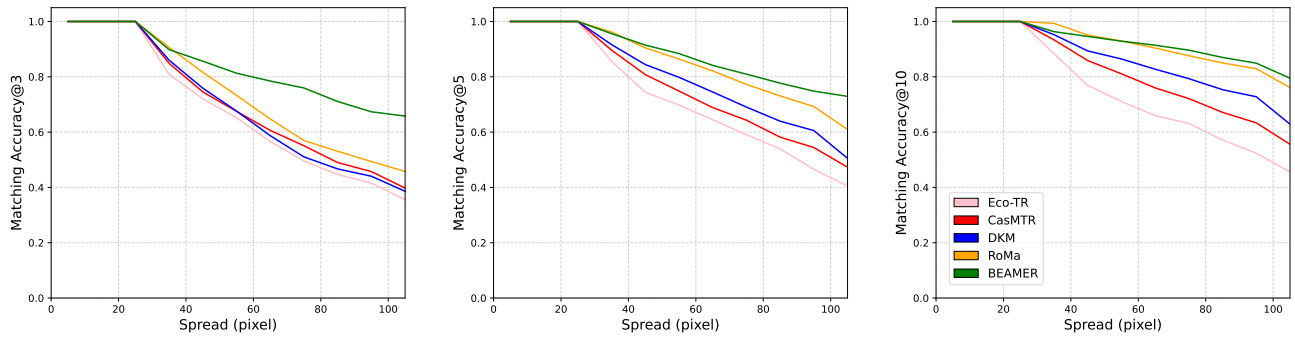

Fig. 4: **Matching accuracy at 3, 5, and 10 pixels on MegaDepth-8-scenes for increasing *spread* levels.** State-of-the-art methods (EcoTR [24], CasMTR [12], DKM [16] and RoMa [19]) degrade significantly when the *spread* increases while BEAMER remains robust by effectively preserving and propagating multiple hypotheses.

$$\sum_{k=1}^{N} \mathcal{L}\left(\mathbf{p}_{\mathrm{s},k}^{\mathrm{GT},1}, \mathbf{p}_{\mathrm{t},k}^{\mathrm{GT},1}\right), \tag{3}$$

where

$$\mathcal{L}\left(\mathbf{p}_{\mathrm{s},k}^{1}, \mathbf{p}_{\mathrm{t},k}^{1}\right) = -\sum_{l=1}^{L} \ln\left(\mathcal{C}_{\mathbf{p}_{\mathrm{s},k}^{l}}^{l}\left(\mathbf{p}_{\mathrm{t},k}^{l}\right)\right), \tag{4}$$

with $\mathbf{p}_{\mathrm{s},k}^{l} = \left\lceil \frac{\mathbf{p}_{\mathrm{s},k}^{1}}{2^{l-1}} \right\rfloor$ and $\mathbf{p}_{\mathrm{t},k}^{l} = \left\lceil \frac{\mathbf{p}_{\mathrm{t},k}^{1}}{2^{l-1}} \right\rfloor$.

3. EXPERIMENTS

3.1. Datasets and Evaluation Protocol

We are interested in evaluating the performance of BEAMER and state-of-the-art methods for several increasing spread levels of the neighboring source locations correspondents. To do so, we consider dense GT correspondences from two datasets. The MegaDepth [25] dataset includes two test sets: the traditional MegaDepth-1500 [5] consisting of 1500 pairs from two scenes, and MegaDepth-8scenes [16], which offers more diversity with 2400 pairs spanning eight different scenes. The HPatches [21] dataset consists of multiple pairs of planar scenes.

Given a pair of source/target images, for each source pixel, we consider a patch of 16x16 pixels (*i.e.* the area covered by the corresponding pixel at scale 1/16) and compute the bounding box of their GT correspondents in the target image. The largest side of the bounding box is called the *spread* (in pixels). Doing so allows us to classify the GT correspondences into increasing spread levels. The evaluation is performed using the matching accuracy [26] at three pixel error thresholds: 3, 5, and 10 pixels. We compare our method against state-of-the-art dense matching techniques that follow different refinement strategies. EcoTR [24] performs coarse-to-fine matching using a sequence of zooms with a local window. CasMTR [12] follows a similar coarse-to-fine approach but applies local window refinement at each scale. DKM [16] and RoMa [19] use regression-based strategies for coarse-to-fine matching. Compared to these methods, BEAMER is the only method that seeks to preserve and propagate multiple hypotheses across scales.

3.2. Experimental Results

Figure 4 shows the matching accuracy at 3, 5, and 10 pixels for several increasing spread levels. We observe that state-of-

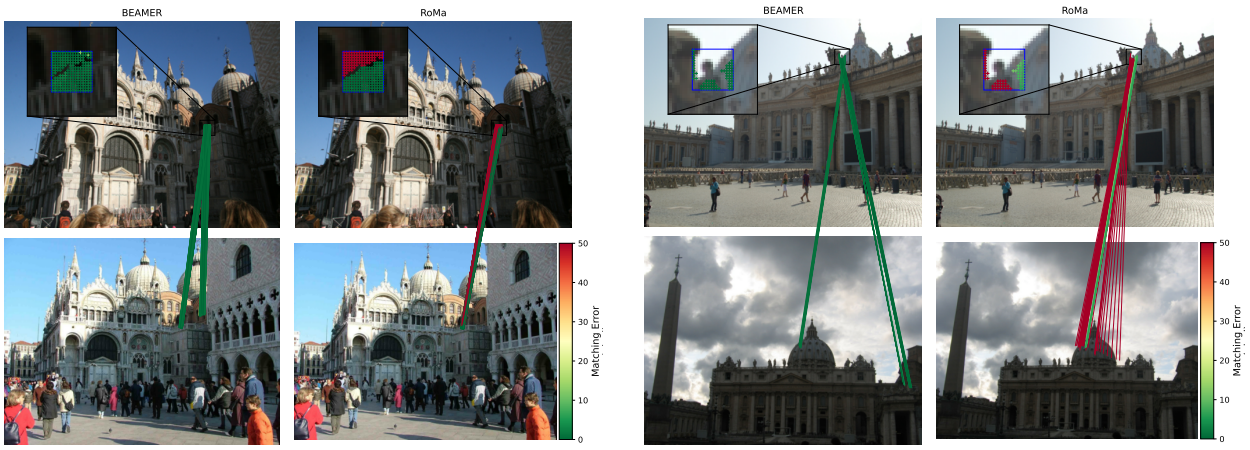

Fig. 5: **Qualitative comparison: we show the correspondents found by BEAMER and RoMa [19] for a 16×16 source patch.** In these examples, the GT correspondents are located on two different modes. Only correspondences with ground truth are displayed and the line color indicates the matching error in pixels. RoMa, which cannot propagate multiple hypotheses across scales, has difficulty finding correspondents, while BEAMER, designed to preserve and propagate multiple hypotheses across scales, successfully identifies the correspondents.

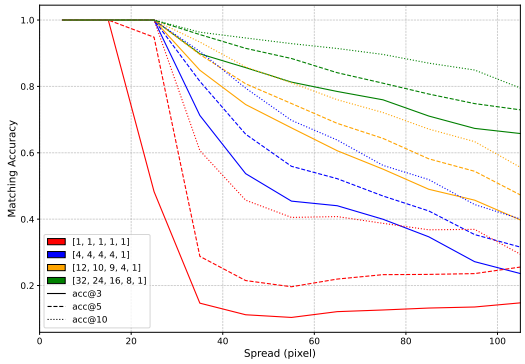

Fig. 6: **Ablation study of the number of hypotheses propagated at each scale [K_5,K_4,K_3,K_2] on MegaDepth-8-scenes.**

the-art methods EcoTR [24], CasMTR [12], DKM [16] and RoMa [19] degrade significantly in accuracy as the spread increases. In contrast, BEAMER's ability to preserve and propagate multiple hypotheses leads to superior accuracy across all settings.

This behavior is illustrated in Fig. 5, where we show the correspondents found by BEAMER and RoMa [19] for a 16×16 source patch. In these examples, the GT correspondents are located on two different modes. RoMa, that is not able to propagate multiple hypotheses across scales, struggles to find the correspondents, whereas BEAMER that was designed to preserve and propagate multiple hypotheses across scales successfully finds the correspondents. More quantitative results can be found in supp. mat.

The quantitative results are summarized in Tab. 1. BEAMER consistently surpasses state-of-the-art methods: as the spreading increases, the difference between BEAMER and the other methods becomes more pronounced.

3.3. Ablation Study

Figure 6 presents an ablation study of the number of hypotheses propagated at each scale [K_5, K_4, K_3, K_2]. We observe that restricting the search to only the most promising hypothesis [1, 1, 1, 1] leads to poor performance. Next, we increase the number of hypotheses to the experimental values from Fig. 2 [12, 10, 9, 4], and we can see that the performance consistently improves. Finally, we show that the hyperparameters chosen for BEAMER [32, 24, 16, 8] significantly improve the results, compensating for the fact that the feature extractor is not perfect.

4. CONCLUSION

In this paper, we introduced BEAMER, a novel coarse-to-fine dense image matching approach that learns to preserve and propagate multiple hypotheses across scales using a beam search strategy. Unlike existing coarse-to-fine methods that propagate a single hypothesis per scale, BEAMER effectively maintains candidate correspondents for each source location at each scale, leading to superior robustness in challenging cases such as depth discontinuities and large viewpoint changes. Our results indicate that BEAMER excels in regions where the correspondents of neighboring source locations are widely spread. Future work could explore incorporating more expressive backbones, such as DINOv2, and optimizing memory efficiency to scale BEAMER to even higher resolutions.

Acknowledgment – This project has received funding from the french ministère de l'Enseignement supérieur, de la Recherche et de l'Innovation. This work was granted access to the HPC resources of IDRIS under the allocation 2022-AD011012858 made by GENCI.

5. REFERENCES

[1] Jared Heinly, Johannes L. Schönberger, Enrique Dunn, and Jan-Michael Frahm, "Reconstructing the World* in Six Days *(as Captured by the Yahoo 100 Million Image Dataset)," in *CVPR*, 2015.

[2] Paul-Edouard Sarlin, Mihai Dusmanu, Johannes L. Schönberger, Pablo Speciale, Lukas Gruber, Viktor Larsson, Ondrej Miksik, and Marc Pollefeys, "LaMAR: Benchmarking Localization and Mapping for Augmented Reality," in *ECCV*, 2022.

[3] 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 *CVPR*, 2018.

[4] Jerome Revaud, Cesar De Souza, Martin Humenberger, and Philippe Weinzaepfel, "R2D2: reliable and repeatable detector and descriptor," *NeurIPS*, 2019.

[5] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *CVPR*, 2020.

[6] Hugo Germain, Guillaume Bourmaud, and Vincent Lepetit, "S2DNet: learning image features for accurate sparse-to-dense matching," in *ECCV*, 2020.

[7] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys, "Lightglue: Local feature matching at light speed," in *ICCV*, 2023.

[8] Johan Edstedt, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg, "DeDoDe: Detect, Don't Describe — Describe, Don't Detect for Local Feature Matching," in *3DV*, 2024.

[9] Guilherme Potje, Felipe Cadar, André Araujo, Renato Martins, and Erickson R. Nascimento, "Xfeat: Accelerated features for lightweight image matching," in *CVPR*, 2024.

[10] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou, "LoFTR: detector-free local feature matching with transformers," in *CVPR*, 2021.

[11] Jiahuan Yu, Jiahao Chang, Jianfeng He, Tianzhu Zhang, Jiyang Yu, and Feng Wu, "Adaptive spot-guided transformer for consistent local feature matching," in *CVPR*, 2023.

[12] Chenjie Cao and Yanwei Fu, "Improving transformer-based image matching by cascaded capturing spatially informative keypoints," in *CVPR*, 2023.

[13] Yifan Wang, Xingyi He, Sida Peng, Dongli Tan, and Xiaowei Zhou, "Efficient LoFTR: Semi-dense local feature matching with sparse-like speed," in *CVPR*, 2024.

[14] Prune Truong, Martin Danelljan, and Radu Timofte, "Glu-net: Global-local universal network for dense flow and correspondences," in *CVPR*, 2020.

[15] Prune Truong, Martin Danelljan, Luc Van Gool, and Radu Timofte, "Learning accurate dense correspondences and when to trust them," in *CVPR*, 2021.

[16] Johan Edstedt, Ioannis Athanasiadis, Mårten Wadenbäck, and Michael Felsberg, "DKM: dense kernelized feature matching for geometry estimation," in *CVPR*, 2023.

[17] Prune Truong, Martin Danelljan, Radu Timofte, and Luc Van Gool, "PDC-Net+: enhanced probabilistic dense correspondence network," *PAMI*, 2023.

[18] Shengjie Zhu and Xiaoming Liu, "PMatch: paired masked image modeling for dense geometric matching," in *CVPR*, 2023.

[19] Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg, "RoMa: Robust Dense Feature Matching," in *CVPR*, 2024.

[20] Gabriele Berton, Riccardo Mereu, Gabriele Trivigno, Carlo Masone, Gabriela Csurka, Torsten Sattler, and Barbara Caputo, "Deep visual geo-localization benchmark," in *CVPR*, 2022.

[21] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk, "HPatches: a benchmark and evaluation of handcrafted and learned local descriptors," in *CVPR*, 2017.

[22] Zichao Zhang, Torsten Sattler, and Davide Scaramuzza, "Reference pose generation for long-term visual localization via learned features and view synthesis," *IJCV*, 2021.

[23] David Furcy and Sven Koenig, "Limited discrepancy beam search," in *IJCAI*, 2005.

[24] Dongli Tan, Jiang-Jiang Liu, Xingyu Chen, Chao Chen, Ruixin Zhang, Yunhang Shen, Shouhong Ding, and Rongrong Ji, "ECO-TR: efficient correspondences finding via coarse-to-fine refinement," in *ECCV*, 2022.

[25] Zhengqi Li and Noah Snavely, "Megadepth: Learning single-view depth prediction from internet photos," in *CVPR*, 2018.

[26] Matthieu Vilain, Rémi Giraud, Hugo Germain, and Guillaume Bourmaud, "Are semi-dense detector-free methods good at matching local features?," in *VISAPP*, 2024.

HANDLING MULTIPLE HYPOTHESES IN COARSE-TO-FINE DENSE IMAGE MATCHING

Supplementary Materials

1. BEAM SEARCH VISUALIZATION

Figure 1 illustrates the behavior of beam search during the coarse-to-fine matching process.

The source location we consider is in $\times$. In the target image, the selected hypotheses at each scale are displayed in +. There are 32 + at $l = 5$ because $K_5 = 32$, 24 + at $l = 4$ because $K_4 = 24$, etc. The red areas (of 4 pixels each) correspond to the search regions at each scale (at $l = 5$ this area is not represented as it is the whole target image). There are 32 red areas at $l = 4$ because $K_{l+1} = K_5 = 32$. These red areas are the pixels used to perform cross-attention with +. At resolutions $l = 5$, 4 and 3, BEAMER effectively explores distant multiple hypotheses. This ensures that plausible correspondents are considered before progressively refining the search. At finer scales ($l = 2, 1$), the resolution is sufficiently detailed to focus only on local regions, enabling BEAMER to accurately identify the correct correspondent.

We also display in blue the pixels selected (in the source image) to perform self-attention with +. At finer resolutions ($l = 1, 2$), BEAMER primarily focuses on regions around the query location. However, at coarser resolutions ($l = 4, 3$), BEAMER also exchanges information with visually similar regions that may introduce ambiguity or regions that may serve as reference points for accurate correspondence estimation.

One important observation from Fig. 1 is that the red and blue areas represent a significantly smaller subset of the entire pixel grid. This highlights the efficiency of beam search, allowing attention mechanisms to operate effectively even at fine resolutions while limiting the computational cost.

For clarity, we visualize a single correspondence path in Fig. 1. However, this process is applied to every pixel in the source image (and every pixel in the target image since BEAMER is bi-directional), ensuring dense matching across the entire image pair.

2. IMPLEMENTATION DETAILS

2.1. Backbone Architecture

The backbone used in BEAMER is a modified version of ResNet18, designed to produce feature maps at every resolutions (1/16, 1/8, 1/4, 1/2, 1). To improve efficiency, we adjust the feature depth at each resolution to the following values: 256 at scale $l = 5$ (res. 1/16), 256 at scale $l = 4$ (res. 1/8), 128 at scale $l = 3$ (res. 1/4), 128 at scale $l = 2$ (res. 1/2), and 64 at scale $l = 1$ (res. 1).

2.2. BEAMER Architecture

For each resolution, different hyperparameters are used in the attention modules. The feature depth varies across scales as in the backbone but is further reduced to reduce the memory footprint: 256 channels at scale $l = 5$, 128 channels at scales $l = 4$ and $l = 3$, 64 channels at scale $l = 2$, and 32 channels at scale $l = 1$. The number of attention heads and their respective sizes are also adapted (self-attention layers and cross-attention layers have the same hyperparameters at each scale): eight heads of size 64 are used at scale $l = 5$, while scales $l = 4$, $l = 3$, and $l = 2$ utilize four heads of size 32. At the finest scale, $l = 1$, two heads of size 32 are employed. In every attention module, the feedforward network is replaced with a two-layer convolutional network with a kernel size of 3, ensuring better local consistency in the learned representations.

As described in the main paper, different numbers of attention modules are used at each scale. Specifically, four dense attention modules are employed at the coarsest scale ($l = 5$), followed by two beam-attention modules at scales $l = 4$ and $l = 3$, and one beam-attention module at scales $l = 2$ and $l = 1$. A more detailed representation of the beam-attention module is provided in Figure 2, which illustrates its structure and the order of operations.

2.3. Training Details

We classically use MegaDepth as training set. Each training batch consists of a single image pair, where the images are resized such that the largest side is 640 pixels. The training pairs are selected as in DKM, *i.e.* such that half of the image pairs have a minimal overlap of 0.01, while the remaining half contains image pairs with a minimal overlap of 0.35 to include easier cases. The backbone is initially trained from scratch for two hours, only on the coarsest resolution, before integrating it into the full model.

The model is trained using mixed precision (`FP16`) to optimize computational efficiency. Additionally, gradient checkpointing is employed to further reduce memory consumption at the cost of increased training time. Training is conducted on four Nvidia V100-16GB GPUs, using a dataset consisting of approximately 1.7 million image pairs. The

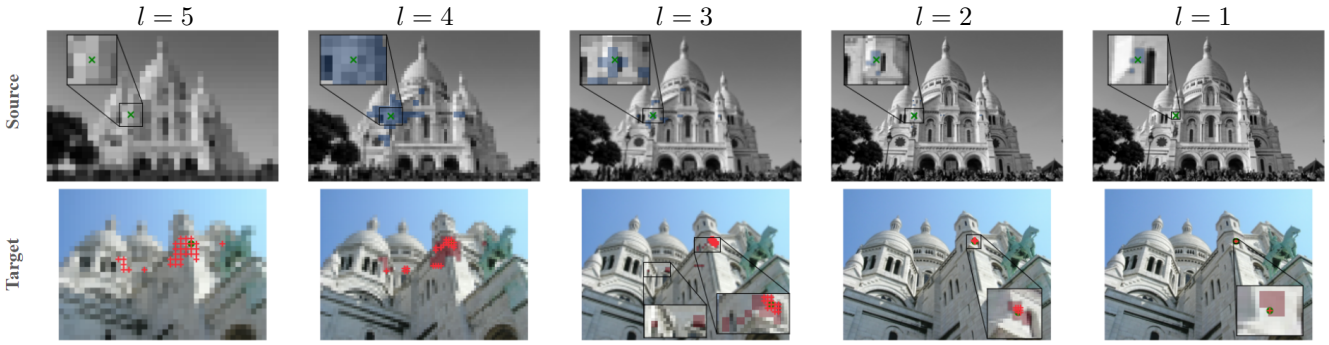

Fig. 1. Visualization of the beam search implemented in BEAMER. See the text for details.

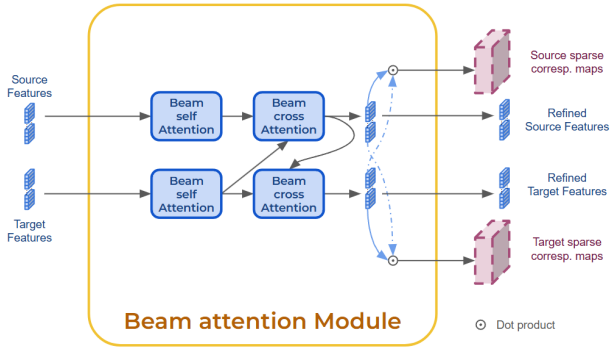

Fig. 2. **Structure of the Beam-attention module.**

learning rate schedule begins with a warm-up phase of 5000 steps, during which the learning rate is linearly increased from 0 to 0.0001, followed by an exponential decay.

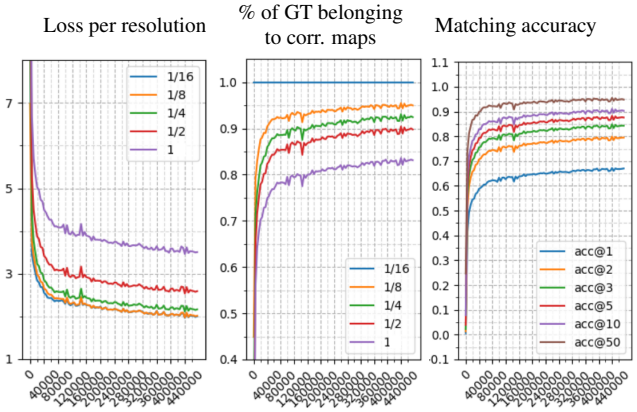

Fig. 3. **Validation metrics during training.**

Figure 3 provides an overview of the key validation metrics tracked during training. In addition to monitoring the loss and matching accuracy at each scale, we also report the percentage of ground-truth correspondences that belong to the sparse correspondence maps, which measures the proportion of cases where BEAMER correctly selects the relevant regions to explore during its beam search. The results indicate that BEAMER progressively learns to propagate the relevant hypotheses across scales, achieving a final accuracy of 1 pixel close to 70%.

3. ADDITIONAL QUALITATIVE COMPARISON

Additional qualitative comparisons are shown in Fig. 4

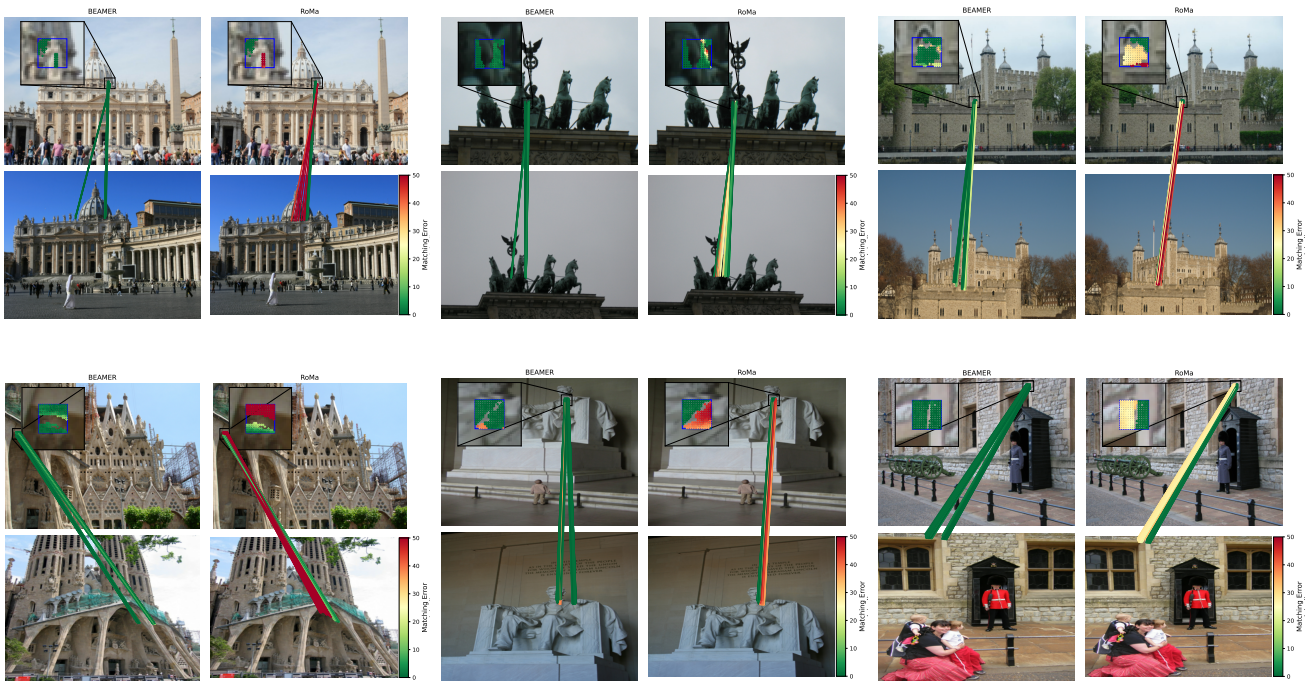

Fig. 4. **Additional qualitative comparison: we show the correspondents found by BEAMER and RoMa for a** 16×16 **source patch.** In these examples, the GT correspondents are located on two different modes. Only correspondences with ground truth are displayed and the line color indicates the matching error in pixels. RoMa, which cannot propagate multiple hypotheses across scales, has difficulty finding correspondents, while BEAMER, designed to preserve and propagate multiple hypotheses across scales, successfully identifies the correspondents.