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 +. 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

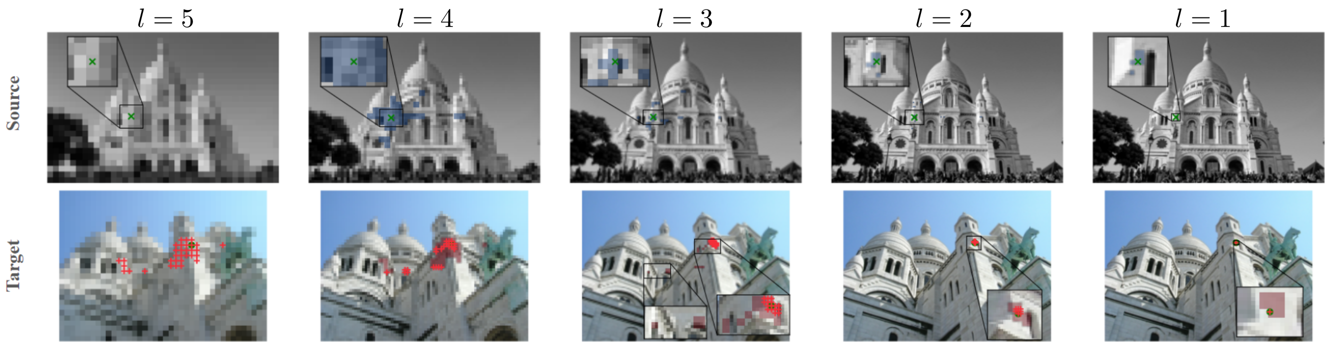

Fig. 1. Visualization of the beam search implemented in BEAMER. See the text for details.

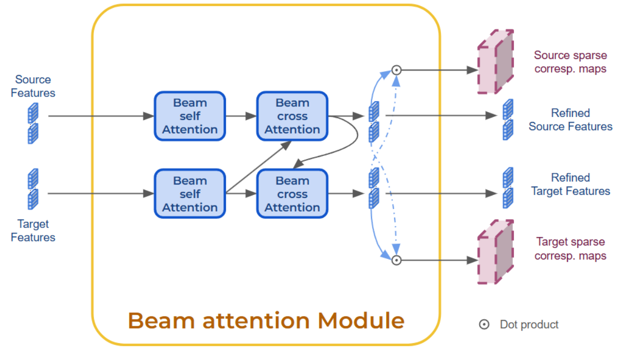

Fig. 2. **Structure of the Beam-attention module.**

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.

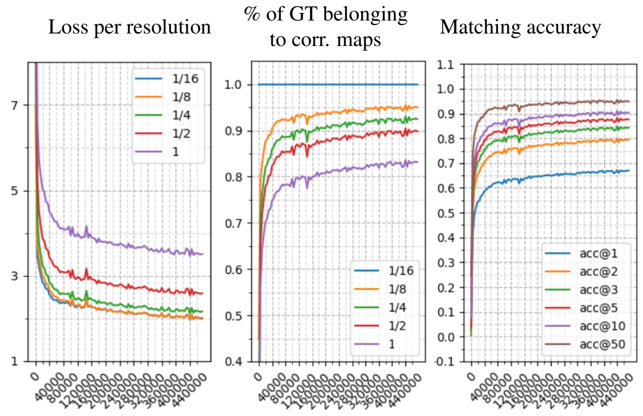

Fig. 3. **Validation metrics during training.**

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

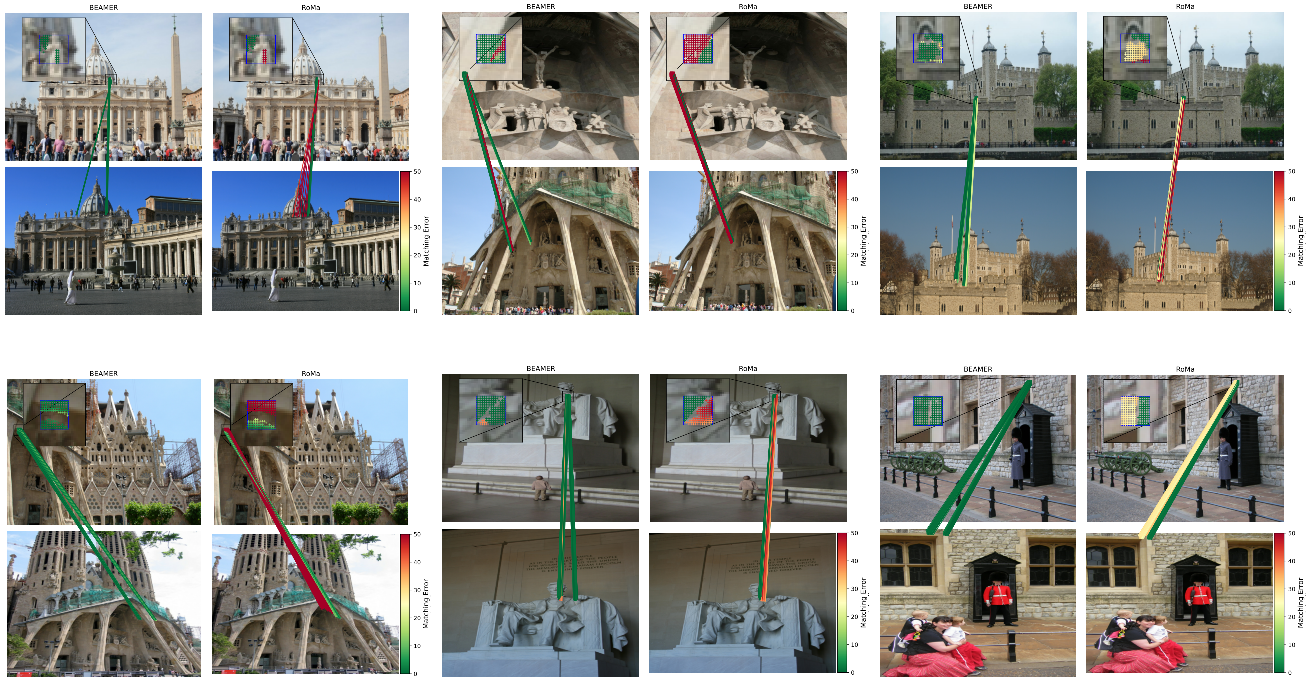

Fig. 4. **Additional qualitative comparison: we show the correspondents found by BEAMER and RoMa for a** 16×16 **source patch.** In these examples, the GT correspondents are located on two different modes. Only correspondences with ground truth are displayed and the line color indicates the matching 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.