[Supplementary Material] Leveraging Semantic Cues from Foundation Vision Models for Enhanced Local Feature Correspondence

Felipe Cadar*[1,2], Guilherme Potje[1], Renato Martins[2,3], Cédric Demonceaux[2,3], and Erickson R. Nascimento[1,4]

[1] Computer Science Department, Universidade Federal de Minas Gerais, Brazil
[2] ICB UMR CNRS 6303, Université de Bourgogne, France
[3] Inria, LORIA, CNRS, Université de Lorraine, France
[4] Microsoft

Project page with code and trained models:
`https://www.verlab.dcc.ufmg.br/descriptors/reasoning_accv24/`

In this supplementary material to the main paper, we provide additional interpretability results of the semantic conditioned features, complementing the qualitative results presented in the paper, more timing statistics, and a deeper analysis on DINO backbone size selection.

1 Inference time analysis and deep-matchers

Table 1. Single Pair Time Analysis. Although we have a longer total time, we can cache the extraction results and reuse them later.

	Time in milliseconds		
	Extraction	Matching	Total
Ours	127.183	1.389	128.571
LightGlue	35.381	26.674	50.657
	Pairs per second		
Ours	7.86	720.03	7.78
LightGlue	28.26	37.49	16.11

For having a better view of the pratical use of our method we included two timing studies:

1) We calculated the extraction and matching time for one pair of images with the ScanNet resolution of $968 \times 1{,}296$ and 2,048 keypoints. In an NVIDIA GeForce RTX 3080/10GB, LightGlue run the feature extraction in 35ms and matching in 27ms, while our method takes 127ms for feature extraction and 1.3ms for feature matching, as shown in Tab. 1

2) To simulate a more realistic scenario, like an SfM pipeline where one image will be matched multiple times, we calculated and cached all keypoints and features for N images and then matched all pairs of images, *i.e.*, $\frac{N(N-1)}{2}$, as usually done in COLMAP. Fig. 1 shows how LightGlue's matching time quickly escalates. Our method becomes more efficient than the deep matcher after just 15 images (a small size dataset).

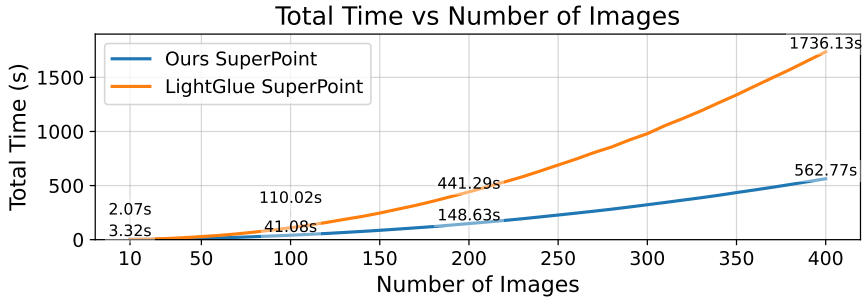

Fig. 1. Time analysis. Time for all images keypoints and descriptors and match all pairs of images. The total time includes GPU-CPU transfers and caching the extractions operations.

We also calculated the compute times to run the 7Scenes benchmark. The timing include some other operations performed by the HLoc framework (commonly used to run the benchmark) that may not be optimized, reducing the gap between the methods. Fig. 2 shows the results.

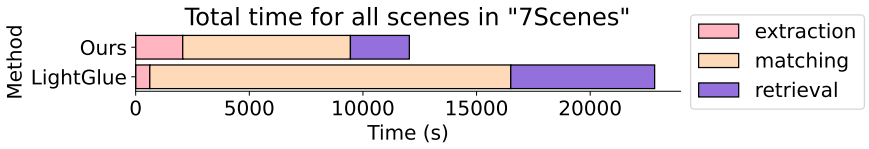

Fig. 2. Time on the 7Scenes benchmark.

2 Further analysis on DINOv2 backbone size.

The sensitivity study on the DINOv2 backbone size reveals that the larger foundation models do not significantly increase pose estimation performance. It is worth noting that ViTs, as used in DINO, are not suitable for fine matching due to the low spatial resolution of their feature maps, but they have strong semantic understanding capabilities due to their global receptive field and attention mechanism. The goal of using foundation models in our pipeline is to provide semantic guidance for the texture features, and the semantic understanding of

small models have shown to be good in our experiments, while reducing computational overhead. This aligns with results reported in the DINOv2 paper, where tasks requiring general semantic understanding such as instance-level recognition and semantic segmentation (DINOv2 paper – Tabs. 7,9,10), exhibit similar performance across all backbone sizes.

3 Interpretability of the features

The proposed description strategy with semantic conditioning also displays interpretable features, which further explain the obtained matching results both quantitatively and from visual inspection (as shown in Fig. 3 of the main paper). We provide more examples of feature interpretability in the following Figs. 3, 4 and 5 of this supplementary material.

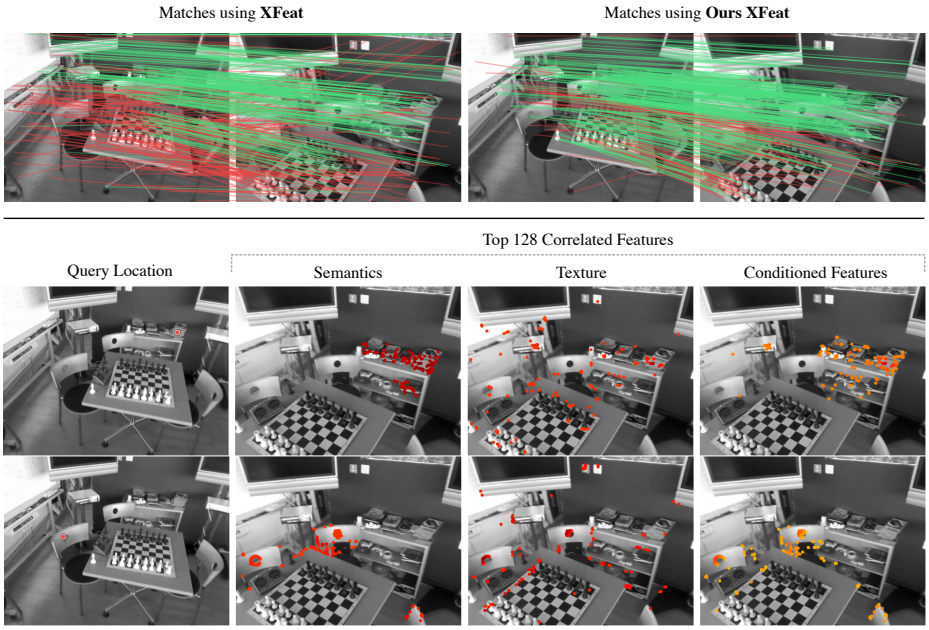

Fig. 3. Interpretability and consistency of the conditioned features. We show the closest 128 matches to a given query keypoint (red point in the first column) for the different descriptors with either sole semantics, the refined texture descriptor, or the proposed semantic conditioned features (fourth column). Hotter colors indicate higher similarity. Notice how the semantic information focuses on instances with similar meaning, like the objects on the table and the chairs, guiding the textural features.

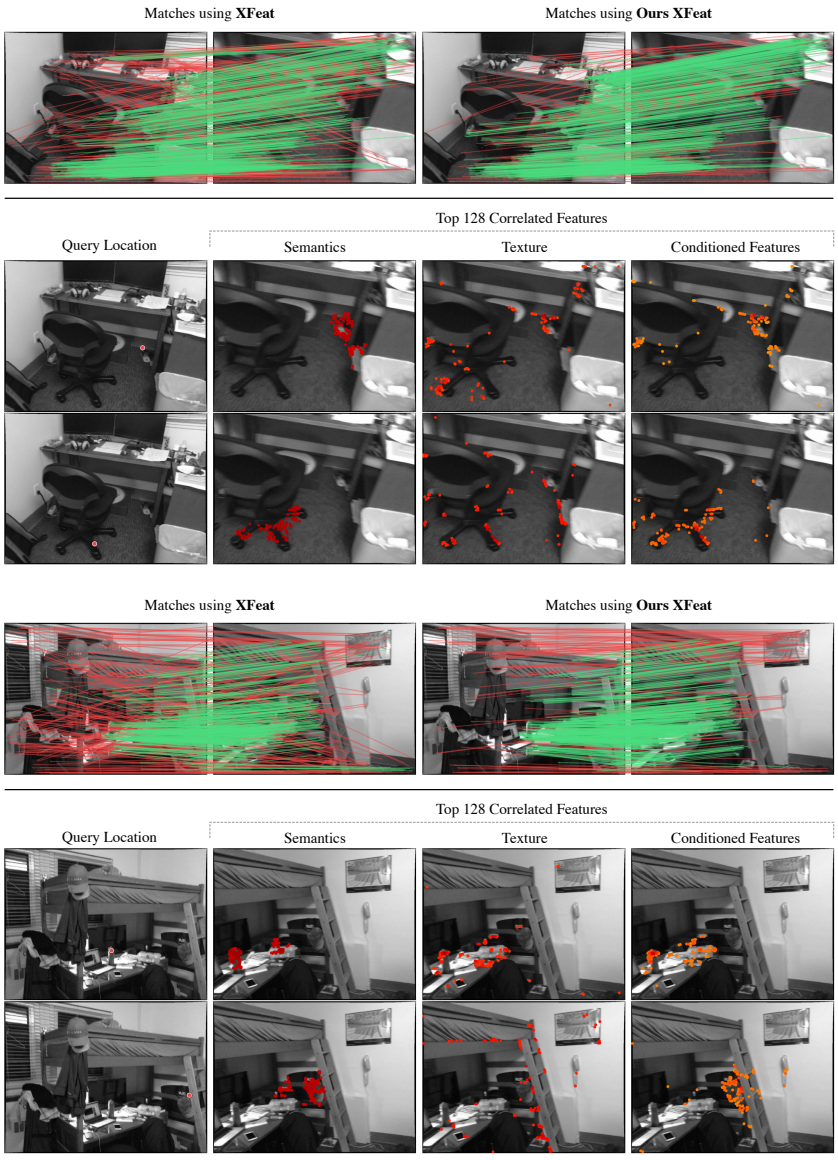

Fig. 4. Interpretability and consistency of the conditioned features. In some cases, the original texture-only features can already show a good matching performance, but they still yield many outliers. In these two examples, the semantic information plays a bigger part in filtering lowering the probability of matching areas with different contexts, also lowering the number of outlier matches.

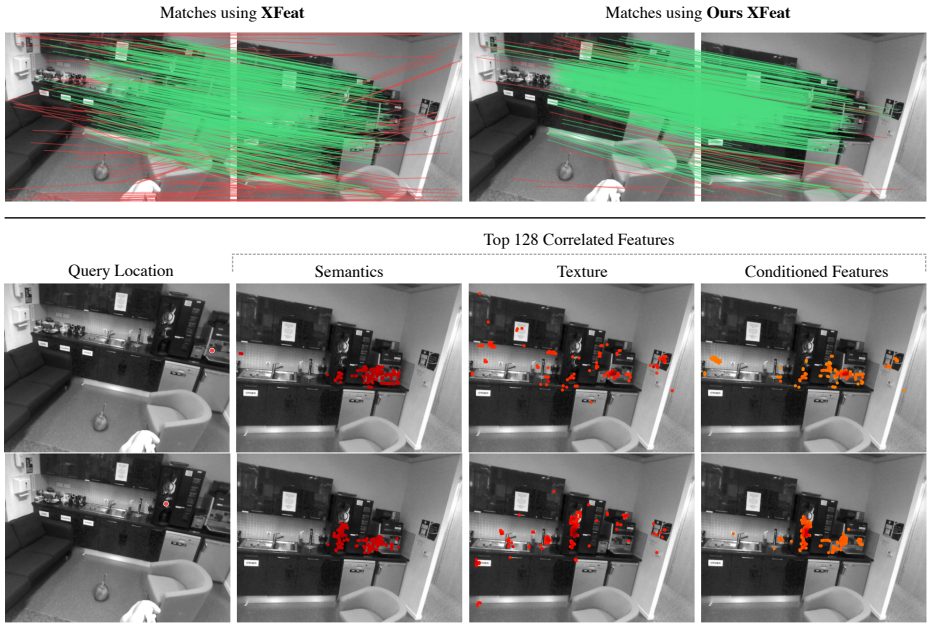

Fig. 5. Interpretability and consistency of the conditioned features. Observe how, with similar semantics in both query points, the texture information helps to differentiate the semantics.