Efficient Descriptor Learning for Large Scale Localization

Antonio Loquercio[1], Marcin Dymczyk[1], Bernhard Zeisl[2], Simon Lynen[2], Igor Gilitschenski[1], Roland Siegwart[1]

***Abstract*— Many robotics and Augmented Reality (AR) systems that use sparse keypoint-based visual maps operate in large and highly repetitive environments, where pose tracking and localization are challenging tasks. Additionally, these systems usually face further challenges, such as limited computational power, or insufficient memory for storing large maps of the entire environment. Thus, developing compact map representations and improving retrieval is of considerable interest for enabling large-scale visual place recognition and loop-closure.**

In this paper, we propose a novel approach to compress descriptors while increasing their discriminability and matchability, based on recent advances in neural networks. At the same time, we target resource-constrained robotics applications in our design choices. The main contributions of this work are twofold. First, we propose a linear projection from descriptor space to a lower-dimensional Euclidean space, based on a novel supervised learning strategy employing a triplet loss. Second, we show the importance of including contextual appearance information to the visual feature in order to improve matching under strong viewpoint, illumination and scene changes. Through detailed experiments on three challenging datasets, we demonstrate significant gains in performance over state-of-the-art methods.

I. INTRODUCTION

Image-based localization aims to accurately estimate and track the pose of a mobile platform with respect to a global map. It is therefore a fundamental task for many robotics and Augmented Reality (AR) applications. The former need localization for path-planning, obstacle avoidance, or manipulation while the latter require high quality pose estimates to correctly project virtual objects into the camera view.

Traditionally, large-scale image-based localization has been treated as an image retrieval problem [1]. However, other approaches, such as [2], achieve higher accuracy using the full 3D model of the workspace to estimate the 6-DoF pose of the query camera employed by the mobile platform. The 3D model is often pre-built with a large set of database images using Structure-from-Motion (SfM) or Simultaneous Localization and Mapping (SLAM) techniques. Recent advances in research now make it possible to construct models on a huge scale consisting of millions of points in only a few hours [3], creating the need for methods that can handle such large datasets.

The key step for image-based localization using 3D models consists of finding correspondences between 2D local features in the query image and 3D points in the model. Commonly, the same type of local image descriptors, *e.g* [4], [5], used to build the 3D model are extracted from the query image. This allows to formulate the correspondence search as an instance of descriptor matching: We select the 3D landmark corresponding to a 2D feature by searching for the nearest neighbor of that feature's descriptor in the space containing 3D landmarks descriptors.

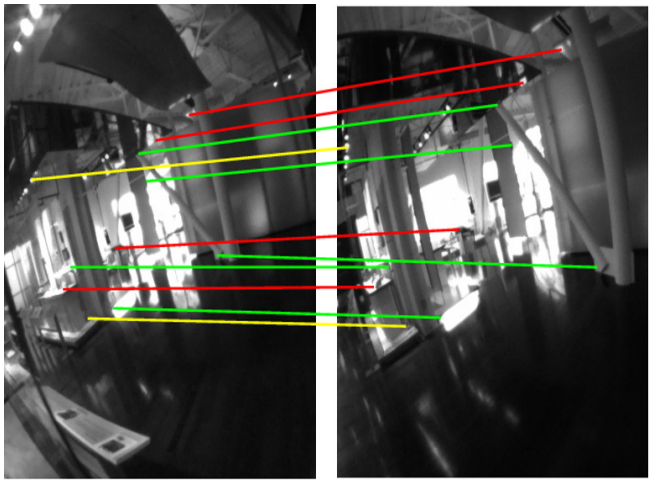


Fig. 1: **Retrieved matches.** The descriptor projection and augmentation methodology herein proposed enables difficult descriptor matches under strong appearance changes. Then, some matches (green) may still be possible without making use of our projection technique, relying on search directly in binary descriptor space. However, the proposed linear projection enables additional, more challenging matches (yellow), that can be further improved by including broader context information (red).

However, as the size of the environment increases, the large-scale search required to establish correspondences is particularly challenging. On the one hand, due to the high dimensionality of image descriptors, not necessarily lying in an Euclidean space, search is computationally expensive [6]. On the other hand, strong illumination, viewpoint, or scene changes, as well as perceptual aliasing [1], can quickly hinder descriptors repeatability and discriminability.

To make retrieval both tractable and efficient, current approaches rely on classical dimensionality reduction techniques [7], or on more elaborate optimization methods aimed to generate discriminative and compact descriptors [8]–[10]. When designing such projection systems, two competing constraints need to be taken into account: Capturing the complexity involved in the search problem and introducing as low latency as possible, to enable real time implementations even on devices with limited computational capabilities. Trading off between these two requirements, we propose in this paper two ways to successfully map descriptors to a

[1]Autonomous System Lab, ETH Zürich
[2]Google Inc., Zürich

lower dimensional Euclidean space while maximizing their matchability and discriminability: a fast linear projection and a more accurate, but slower, context-augmented mapping. The resulting descriptors, learning an invariance to strong changes in appearance, enable retrieval of hard matches as the ones shown in Fig. 1. Overall, this work makes the following contributions:

- We use a novel supervised learning strategy to train an efficient linear projection of descriptors.
- We show that including context information in the projection function consistently increases the retrieval performance, outperforming even completely learned descriptors.
- We demonstrate the effectiveness of our methods via an extensive experimental evaluation, and benchmark the performance gains against baseline methods.

II. RELATED WORK

Great improvements in designing image-based localization systems have been achieved in the last few years [2], [11]–[14]. All of them rely on some form of direct or indirect matching between database and query descriptors. In this paper, we concentrate on how to reduce descriptors dimensionality in order to decrease memory requirements, speed up the matching process, and increase its efficiency.

Traditionally, classical approaches from statistics are used to encode descriptors while maximizing their information content. The most commonly used linear technique, principal component analysis [7], performs a mapping to the hyperplane maximizing data variance. Other techniques try to generalize this idea to non-linear mappings by either using the kernel trick [15] or with graph based methods [16]–[18]. However, such *unsupervised* projections are not tuned on the final matching task.

Instead, all relevant *supervised* learning based approaches fall into one or both of the following two categories: (i) learn a metric suited to the classification problem [19], [20] or (ii) learn a projection function of the input data [8], [9], [12], [21]. In this work, we set the projected space to be Euclidean, so that the resulting descriptors, avoiding all the complexities related to a learned metric, can be easily used within existing search algorithms.

The spectrum of possibilities to learn a projection function ranges from employing statistical tests [12], convex-optimization [21], or minimizing a margin based loss [8], [9]. However, learning a mapping to a space where nearest neighbor search is efficient is not a trivial task. In fact, the notion of a nearest neighbor depends on the interaction between all other points. Therefore great care has to be taken in order to present to the optimization the right training data for learning these relationships. For this reason, *supervised* methods usually require large amount of training data to generalize. Moreover Philbin *et al.* proposed in [8] to directly include in the optimized loss function *hard* pairs $\{(d_i, d_j)\}$, where d_i and d_j lie very close in input space despite representing non matching 3D points. While this choice proves to boost the optimization performance, it still misses that hard cases in input space might not be hard cases in projected space, and vice-versa. This is particularly true when the learned projection function is not linear. To face this issue, we present a novel learning policy that, concentrating on the current hard cases in projected space, allows to drastically reduce the amount of training data required to generalize.

Nonetheless, low-level features as corners, edges, or blobs are known to have limited discriminability in large environments with high repetitive structures, where many detected features look alike. State-of-the-art approaches try to tackle this issue by learning more powerful features [10], [21]–[24]. Many of these works using convolutional neural networks (CNNs) have reported promising results, yielding an edge over hand-crafted solutions as SIFT [4] or FREAK [5].

In this paper, we will generalize the techniques mentioned above to encode any given type of image patch descriptor in a compact Euclidean space while increasing discriminability. With our results, establishing the 2D-3D matches required by image-based localization algorithms is both reliable and efficient, even in challenging environments.

III. METHODOLOGY

Our aim is to learn a projection function f, parametrized by θ, mapping an image patch descriptor $d \in X$ to a lower dimensional Euclidean space $T \in \mathbb{R}^d$:

$$f_\theta : X \mapsto T \tag{1}$$

The small dimensionality of the projected descriptors $f_\theta(d)$ will both reduce the memory footprint and increase the efficiency of k-Nearest Neighbor (k-NN) search enabling the use of fast algorithms such as KD-Trees [25]. Moreover, since the projected space is Euclidean, we can use the L_2 distance to compare descriptors and determine their similarity. In general, we would like f_θ to introduce as small overhead as possible to enable real time implementations on devices with limited computational resources.

Building on the success of state-of-the-art supervised learning systems [8], [9], [12], [21], we model projection as an optimization problem. This is typically done by optimizing a loss L on the projection function f_θ. The main objective of the optimization is to decrease the distance in projected space between matching descriptors while increasing the distance between non-matching ones [10].

A. Linear descriptor projection

The case where f_θ is a real valued matrix W, which is equivalent to a linear projection, is of crucial importance for all devices with limited computational resources. To learn the components of W, we optimize a triplet loss function L, proved in [26] to optimally model relative interactions between input samples. In addition, a novel learning policy is used in order to increase performance while decreasing training time.

Triplet loss. A triplet is defined as a collection of 3 descriptors (X_1, X_2, X_3) where X_1 and X_2 are matching, while

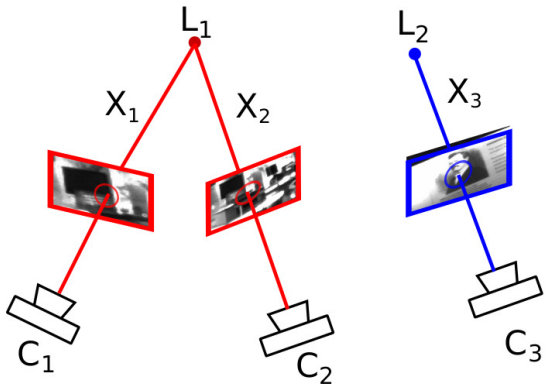


Fig. 2: **A descriptor triplet** (X_1, X_2, X_3). The first two components of the triplet, X_1 and X_2, are matching, meaning they observe the same feature. X_3, on the other hand, is not matching, because it describes a different interest point.

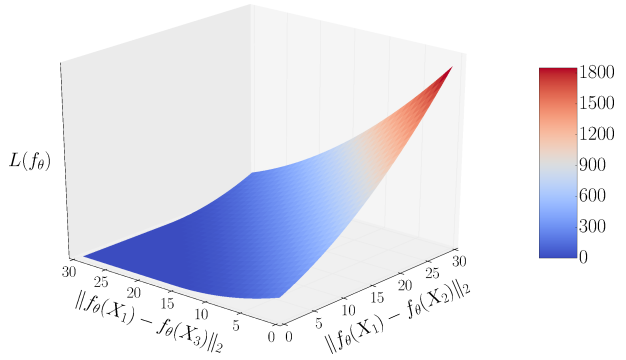


Fig. 3: **Margin based loss.** Equation 3 is plotted in this illustration as function of the L_2 distance between matching and non matching samples. Even if convex in the latter parameters, $L(f_\theta)$ is not convex in the projection function parameters θ.

X_3 is not, as depicted in Fig. 2. We define $d(.)$ as the L_2 distance between two mapped samples, that is:

$$d(X_i, X_j) = \|f_\theta(X_i) - f_\theta(X_j)\|_2 \tag{2}$$

Accordingly, we optimize a non-convex problem defined by the sum of a projection loss L_p and a L_2 regularizer on the function parameters:

$$L(f_\theta) = L_p(f_\theta) + \lambda \|\theta\|^2 \tag{3}$$

Where the projection loss is defined as:

$$L_P(f_\theta) = \sum_{triplets} \max(d(X_1, X_2) + margin - d(X_1, X_3), 0)^2 \tag{4}$$

The latter, depicted in Fig. 3, aims to set the L_2 distance between non matching samples at least a *margin* larger than matching ones' distance. Even if such a kind of L_2 loss is known to be sensitive to outliers [27], the high accuracy of our data allows us to properly generalize [2].

Training cycle. Inspired by boosting, multiple works reported the benefits of a hard negative mining policy, where the optimization concentrates on samples with higher loss [8], [22], [23]. However, to efficiently capture difficult cases without pre-computing them, we herein propose to generate them on the fly. In this way, we can point the optimization's attention toward the wrongly classified pairs. Thus initializing the training with only a limited amount of input data suffices, since during learning we will generate new *hard* triplets. Those are generated so that X_3 is one of the nearest neighbors of X_1 in projected space, even though it should not be matched.

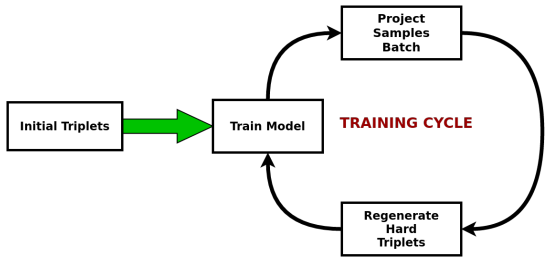
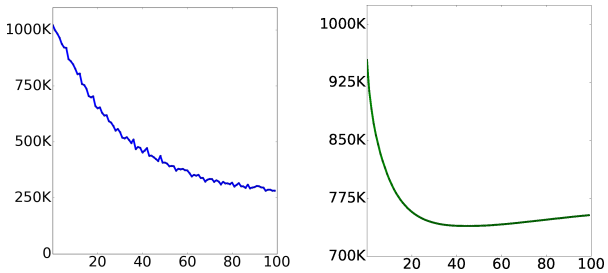


Fig. 4: **Training cycle of the proposed learning method**. To initialize the training, some initial samples triplets should be provided. After these have been back-propagated, we project a batch of training samples, and use them to produce new *hard* triplets. These will be used to continue training the model.



(a) Train Loss per Epoch (b) Validation Loss per Epoch

Fig. 5: **Online cross-validation**. The learned parameters are periodically evaluated against a validation set unseen during training. The parameters with lower validation loss will be picked-up for testing. This procedure, alongside L_2 normalization, reduces over-fitting on the training set.

It should be noted, however, that all boosting-inspired policies can work properly only under the assumption of highly accurate training sample's labels [28]. Otherwise, the algorithm would spend too much "effort" on correctly classifying noisy examples, resulting in poor performance.

Our cyclic learning process, depicted in Fig. 4, drastically reduces training times while increasing performance. To avoid over-fitting, we used an L_2 regularizer on θ. Additionally, we cross-validate the learned parameters periodically, as shown in Fig. 5.

Given that we try to maximally separate non matching points by a linear projection, the optimization goal is equivalent to learning a Mahalanobis matrix $M = W^T W$. However, by learning W directly we avoid all the complications related to adding the positive semidefiniteness constraints on M. Furthermore, as in multi-class LDA [29] we want to project data on the subspace which contains the most class variability. However, as the assumptions of independence and normal distribution of the input variables do not hold, a poor performance can be expected from LDA. Indeed, in our early experiments we noticed an almost 11% performance drops w.r.t. the baseline method, *PCA*.

A similar learning problem was already tackled in [8], [21]. Even though our approach is conceptually similar to theirs, the use of triplet loss and our training policy allow us to increase performance, while controlling (in contrast to [21]) the output dimensionality. Indeed, in the experimental section below we will show that the resulting linear projection outperforms non-linear methods and successfully generalizes to unseen data.

B. Context augmented descriptors

A problem that is well known in large scale localization is the degradation of performance that inevitably happens in large environments with highly repetitive structures, such as warehouses, offices, and museums. In these conditions low level features are not discriminative enough, and despite the improvement that can be obtained by learning an informative projection f_θ [8], the problem remains essentially unsolved.

However, this can be mitigated by including visual context from the feature neighborhood which additionally improves discriminability. Formally, this is equivalent to changing the projection function to $f_\theta(d_i, I_i)$, taking as input both the original descriptor d_i and its surrounding raw image context I_i. The intuition here is based on the fact that similar non-matching descriptors can be better distinguished if we include additional information coming from their neighborhood.

To achieve this, we need to first properly define context, then find a way to represent it and eventually propose a method to learn a combination between the feature and the context descriptors.

What is context? We define context for a visual feature by a square area around it in the image plane. Examples of context can be seen in Fig. 6. The neighborhood N is tied to the feature detected scale s by the formula:

$$N = C \times s \tag{5}$$

Where C a learned constant selected by cross-validation, as shown in Table I. This way, the context region automatically inherits the scale invariance of the feature detector.

Essentially, context is an image patch that comes from a larger area than the patch used for extracting the descriptor. Therefore, the obvious question is why to use the original descriptor at all. Later in the experimental section, we show that a combination of context and the original feature outperforms an approach that learns only on the context window, and provide a possible reasoning.

C Multiplier	5	10	20	50	100	300
PR AUC	0.418	0.544	0.656	**0.684**	0.670	0.533

TABLE I: **Multiplier cross-validation.** To select a constant multiplier of the feature scale, we cross-validated its value using the Precision Recall Area Under the Curve (*PR AUC*). More details about this score will be given in section IV-A. As expected from a qualitative analysis (Fig. 6), $C = 50$ gives the best result.

How to represent context? We want to find a function $g_\rho(I_i)$ that takes as input an image patch I_i and produces a discriminative descriptor of it. Traditionally, this has been solved by aggregating the image features in a single vector such as bag-of-visual words [30], VLAD [31], or Fisher vectors [32]. However, in the last few years convolutional neural networks (CNNs) emerged as a powerful image representation tool [22]–[24], [33]–[35].

We represent $g_\rho(.)$ through the simple architecture presented in [23], generating a 128-float context descriptor. This choice was motivated by the high performance, and the small network size that allows real time evaluation. Similarly to descriptor projection, we train the context descriptor minimizing a triplet loss defined on (I_1, I_2, I_3), where the pair (I_1, I_2) is matching while (I_1, I_3) is not. Depicted in Fig. 7, this is defined as:

$$L(g_\rho) = \sum_{triplets} d(I_1, I_2) + \max(margin - d(I_1, I_3), 0) + \lambda \|\rho\|^2 \tag{6}$$

Where $\|\rho\|^2$ is a regularizer on the CNN weights and $d(.)$ represents the L_2 distance between context descriptors:

$$d(I_i, I_j) = \|g_\rho(I_i) - g_\rho(I_j)\|_2 \tag{7}$$

Note the differences between the losses in Equation 6 and Equation 3. First, in contrast to the latter, the former is not squared. Next, Equation 6 explicitly enforces corresponding patches to be as close as possible in projected space, whereas the loss in Equation 3 only imposes this constraint implicitly. The latter is indeed zero as long as the non matching points' distance is large enough. These differences aim to make our CNN descriptor more robust to photometric changes and affine transformation in the input image space [23]. As expected, our experiments confirmed that losses *ad-hoc* for each task outperform a shared one.

Using descriptor to image correspondences, we can convert the training data we used before to learn a linear projection to image patch triplets, automatically generating training data for this task. Moreover, to again speed up the training process and increase its performance, we employ our hard triplet regeneration scheme described in Fig. 4. The only pre-processing applied to the image patches is the subtraction of the overall training set mean.

How to combine feature and context descriptor? To reduce the computational overhead, we adopted a simple approach: we first stack the original descriptor and the context feature, then we learn a linear projection W_c for the combined feature as in III-A. In the mathematical formalism we used before, this is equivalent to the final projection function:

$$f_\theta(d_i, I_i) = W_c \begin{bmatrix} d_i & g_\rho(I_i) \end{bmatrix} \tag{8}$$

Despite its simplicity, this method proved to learn efficiently a complex synergy between its two components.

Even if a joint training of W_c and $g_\rho(.)$ is possible, we experimentally noticed a slight improvement in performance by learning these two parts separately. The most likely reason is that, with separate learning, we can train each component on a more suited loss. Next, we can present to the respective optimization procedures more suitable difficult cases through our *hard* triplets regeneration policy.

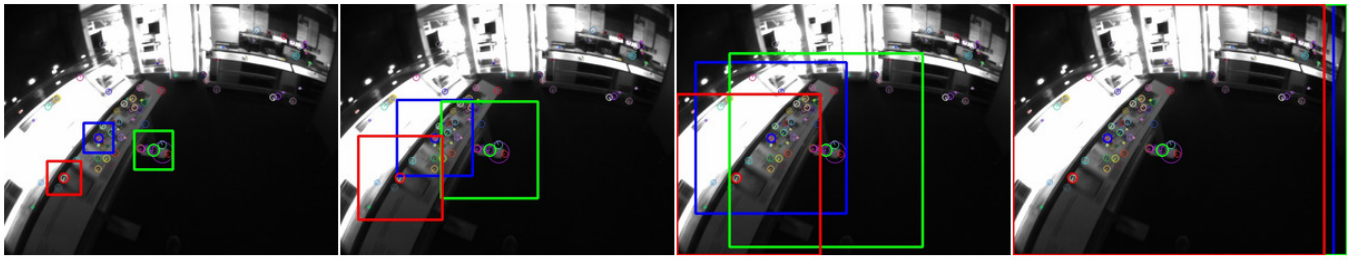


Fig. 6: **Example of local context**. To achieve scale invariance, we couple the context with the feature scale through a constant multiplier. In the figure, the multipliers are respectively $20, 50, 100, 300$. Note that from multiplier 100 boundary effects starts being significant, while for 20 the context captures only few others interesting points. The results in Table I report the multiplier 50 to be the best in term of performance as we would expect from a qualitative analysis.

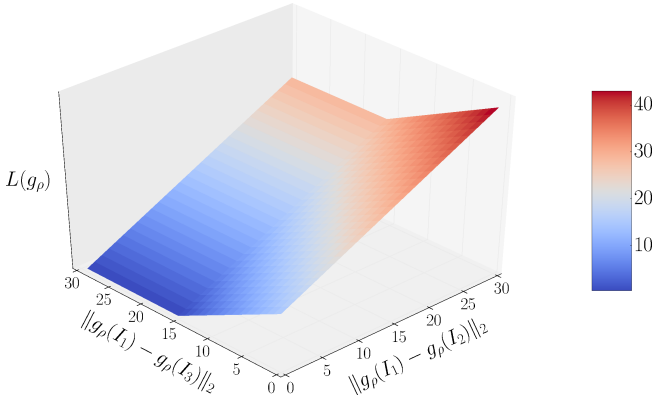


Fig. 7: **Context descriptor loss.** Equation 6 is plotted in this illustration with respect to the L_2 distance between matching and non matching image patches descriptors. Differently from the loss in Fig. 3, we put more emphasis here on the distance between matching image patches, required to be as small as possible. Enforcing this constraint, even if beneficial for learning an image patch feature, results in performance drops if used for training a linear projection.

Full frame context augmentation. To evaluate whether local context performs better than full image context, we used as image descriptor either VLAD [31] or the last hidden layer of some CNNs architecture pre-trained on the object recognition task [33], [36]–[38]. Then we learned a linear combination of the full image descriptor and feature descriptor as presented in the previous subsection. However, as full frame context gets sensitive to large viewpoint changes, a relative performance drop w.r.t. local image neighborhoods is expected. This intuition was confirmed in our experiments.

IV. EXPERIMENTAL SETUP

A. Evaluation methodology

Visual maps used for localization consist of a sparse set of 3D points (a.k.a. landmarks), usually built in advance with SfM. This algorithm reconstructs a 3D scene point from multiple 2D observations. Accordingly, each 3D landmark can be associated to the descriptors describing its appearance in the images it was observed in. The aim of our system is to represent the semantic distance between descriptors by the Euclidean one. Hence, we want projected descriptors corresponding to the same point being closer than non corresponding ones.

To quantitatively evaluate the performance of our proposed methods, we cast the localization problem as an instance of classification. Each landmark represents a class, and at test time we associate new descriptors to them using Nearest Neighbor (NN) search in projected space. This is actually a common practice in the first steps of any localization pipeline [12].

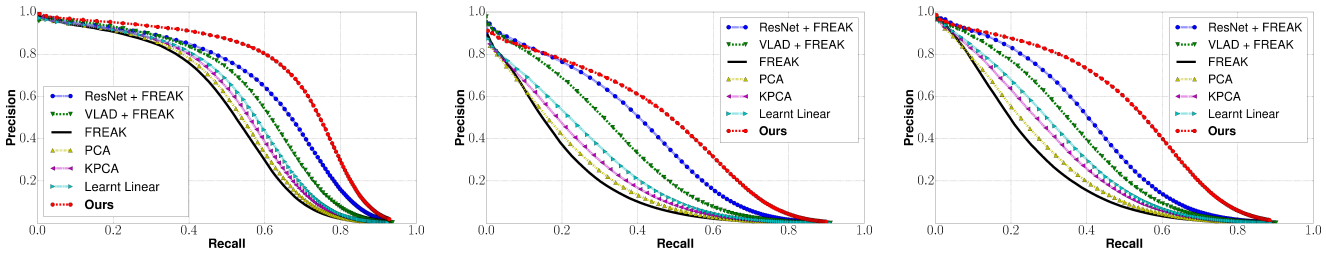
More specifically, for each landmark in a map, we use one of the associated descriptors as query, and then retrieve the set of descriptors with an Euclidean distance lower than a predefined threshold. Out of this set, the ones belonging to the same landmark are considered to be true positives, whereas the ones that are not are false positives. This setup allows to calculate precision-recall pairs (PRs) for each landmark, that are eventually averaged to compute the final classification's PR. For consistency, we used for evaluation only landmarks with a predefined number of associated descriptors. In conclusion, we use a set of thresholds to compute a PR curve, which is usually summarized by its Area Under the Curve (AUC).

We believe this methodology can better evaluate the impact of our solutions compared to evaluating the performance of the full localization task, where higher-level algorithms can distort the influence our our system. In fact, we are primarily interested in testing the quality of our projection function, whose applications are not limited to localization.

To generate training data for our algorithms, we use the 3D model of a large museum built with SfM. Given the size and repetitiveness of this environment, we aim to learn and later test invariance to strong appearance changes and perceptual aliasing. Our framework employs 512-bit FREAK [5] descriptors. The train, validation, and test datasets are obtained by selecting different and not intersecting regions of the environment. As typical in any machine learning pipeline, the train and validation splits are used at learning time, the former to actually perform the optimization and the latter to check for over-fitting and pick the best model. The test set is solely used for evaluation. To further prove generalization on visually different data, we test our approaches on two other indoor 3D models recorded in a large office building. Table III presents more details about these datasets.

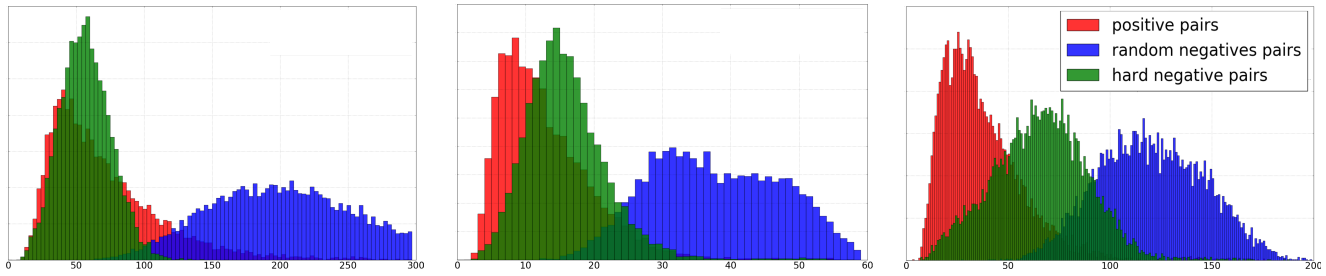
Test	FREAK (512b)	PCA (16d)	KPCA (16d)	Learned Linear (16d)	Full Frame VLAD + FREAK (16d)	Full Frame ResNet + FREAK (16d)	Context Augmented Descriptor (Ours) (16d)
Museum	0.488	0.495	0.512	0.531	0.573	0.612	**0.684**
Office1	0.178	0.188	0.210	0.237	0.310	0.379	**0.433**
Office2	0.228	0.244	0.277	0.301	0.348	0.394	**0.505**

TABLE II: **PR Area Under the Curve (AUC).** Generalization results over 3 evaluation datasets. All projection methods are learned with the train split of the Museum dataset. The corresponding PR curves are depicted in Fig. 8



(a) Museum Dataset (b) Office1 Dataset (c) Office2 Dataset

Fig. 8: **PR curves.** Despite being learned on the train split of the Museum dataset, our methods generalize very well to the visually different data of the Office datasets. Indeed, it can be observed that both our linear and context augmented projection consistently outperform all baseline approaches. Interestingly, search with FREAK descriptors performs very poorly. We believe this is due to the strong changes in appearance present in our datasets. The corresponding PR AUC are presented in Table II. All projected descriptors, except the original FREAK, are 16 dimensional floats.



(a) FREAK (512b) (b) Learned linear (16d) (c) Learned with context (16d)

Fig. 9: **Pairwise distances distribution.** In large and repetitive environments it is difficult to discriminate between matching (in red) and non-matching pairs (in green and blue). In fact, for each descriptor there are going to be many non-corresponding nearest neighbors (distances in green). Our linear and context projection alleviates this issue, improving descriptors discriminability.

Dataset	Notes	Descriptors	3D Landmarks	Keyframes
Museum	Train	2.7M	420K	11.4K
	Validation	30K	2.8K	0.4K
	Test	0.13M	19K	0.6K
Office1	Test only	0.23M	37K	3K
Office2	Test only	0.11M	15K	2.1K

TABLE III: **Train and evaluation datasets.** Only the Museum dataset is used for training our models. To test generalization on visually different data, unseen during training, we evaluate on two other challenging indoor datasets.

B. Optimization

We train our methods using the Google Tensorflow framework, optimizing our loss with Adagrad [39] and default parameters. In all our training algorithms, we set the batch size to be 1K, and employ an exponential learning rate decay with coefficient 0.8 for each epoch. Our learning policy allows us to get good results already after as few as 10 epochs. The training times are around 30 minutes for linear projection and approximately 2 hours for the CNN context projection.

V. RESULTS

We benchmark our models only against traditional baseline methods such as PCA [7] or KPCA [15], because their performance was superior in our datasets than any other learned mapping [8], [9], [12]. Fig. 10 shows the results of the linear and a context augmented projection at different dimensions. As expected, higher dimensionality entails a gain in performance, at the cost of increased memory footprint and computational complexity. We believe 16 dimensions represented by single-precision floating point numbers to offer optimal trade-off between performance and complexity, because it unlocks high quality and fast k-NN search, not raising the memory requirements w.r.t. 512-bit FREAK.

From Table II we can conclude that our proposed linear projection not only outperforms any other considered context-free method, but also generalizes very well to visually different data of the Office datasets. In fact, in those environments it increases the PR AUC score of more than 30% w.r.t. FREAK descriptors and 13% w.r.t. our strongest baseline, *KPCA*.

Context augmented descriptors, described in sec. III-B, provided impressive results: On the challenging Office1 datasets, their use resulted in a PR AUC score of 0.433, more than twice as high compared to using *KPCA* (0.210) or unprojected binary descriptors (0.178).

Results in Fig. 8 and Table II show that local context is more informative than full frame context, represented either by CNNs or VLAD. This can be justified by the observation that the same 3D landmark can appear in very different images due to large viewpoint changes. For the sake of brevity, we only show the performance obtained with Residual Net [33], because it outperformed all other CNN architectures.

What might be surprising is that FREAK descriptors are outperformed by any projection in all datasets. The intuition behind this result is that as our datasets are very challenging in terms of appearance changes and perceptual aliasing, they cause a lot of noise in the binary descriptors. Thus, methods such as *PCA* can already beat their performance by dropping low variance dimensions in data.

Our evaluation confirms two important hypothesis of this paper:

1) Our linear approach is able to learn already a powerful projection function of input descriptors. Being a linear mapping, it also remains highly computationally efficient.
2) Context information can substantially increase the discriminability power of low level features as edges, corners, or blobs (Fig. 9).

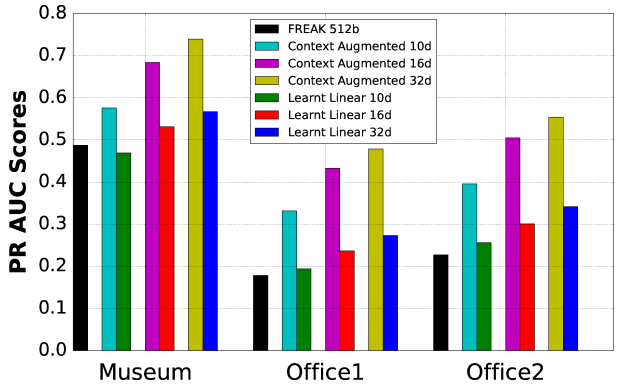


Fig. 10: **Influence of descriptor dimensionality.** Comparison of learned linear and local context augmented projections over different dimensionalities.

A. Why not only learning?

Fig. 11 shows that the learned combination of the context descriptor and the binary descriptor actually outperforms any of the two components considered individually, despite lying in a much smaller-dimensional space. This proves that a complex synergy between the visual feature and its context is actually learned, but it also poses the question of why the CNN architecture does not learn automatically this interaction between the center of the image patch (from where the binary descriptor is extracted) and its neighborhood.

Our main hypothesis is that since the CNN architecture has access to an image patch much wider than the original feature, it produces a descriptor more sensitive to large viewpoint changes. Surprisingly, and despite its simplicity, our proposed approach can learn which kind of information extracted from the CNN is actually beneficial for retrieval.

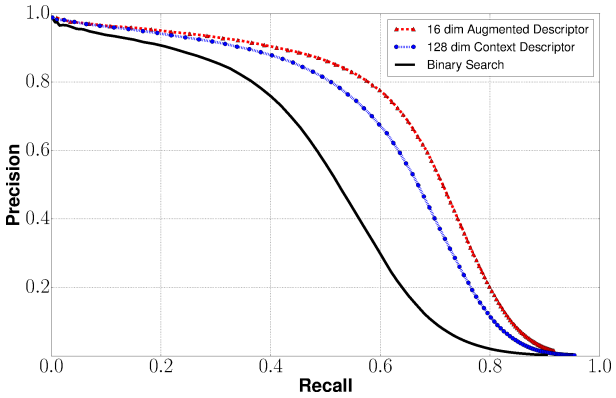


Fig. 11: **Coupling Learned:** The combination between context and visual feature beats the performance of the two parts considered individually, despite its smaller dimensionality. This proves the suggested approach yields better results than only using a single (handcrafted or learned) image patch descriptor.

Overall, we believe that the handcrafted descriptor's role could eventually be substituted by a fully learned approach, either relaxing some constraints, *e.g.* linear projection, or employing more training data. Hence, future work should both integrate our methodology to end-to-end learned descriptors [22], and test its validity on several other sensor types.

B. Computational costs.

Table IV provides the timing of our projection methods. Our GPU variant ran on a Nvidia GeForce GT 730M, while our CPU version ran on an Intel i7-core 2.50GHz processor. Our approach can still be optimized, particularly for dense computations.

Assuming to use a limited number of features for query image, *e.g* 100, we could already run a localization pipeline at around 10Hz [40].

	GPU (μs)	CPU (μs)
Linear Projection	4 ± 0.8	2.8 ± 0.9
Context augmented projection	595 ± 2	1144 ± 1.8

TABLE IV: **Computational cost** for one descriptor (in batch)

VI. CONCLUSIONS

In this paper, we have presented a novel approach for learning a linear and a context augmented feature projection. Even taken on its own, the linear projection beats in term of performance and computations more burdensome non-linear projection approaches such as Kernel PCA for nearest neighbor search in descriptor space. We further improved upon these results by additionally proposing a method to include context information to establish hard matches under strong

appearance changes. Altogether, this resulted in a significant performance gain yielding over 50% higher (compared to binary search in original binary descriptor space) precision at the same recall and thus outperforming any other state-of-the-art linear or non-linear projection method.

While motivated by a scenario involving visual maps, the proposed methodology is of interest to a much wider audience as it can not only be applied to other map and descriptor types (e.g. LiDAR based maps) but can also be understood as a general technique for improving performance of numerous matching and retrieval tasks.

REFERENCES

[1] M. Cummins and P. Newman, "Fab-map: Probabilistic localization and mapping in the space of appearance," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.

[2] T. Sattler, B. Leibe, and L. Kobbelt, "Fast image-based localization using direct 2d-to-3d matching," in *ICCV*. IEEE, 2011, pp. 667–674.

[3] J.-M. Frahm and others (19 authors), "Building rome on a cloudless day," in *European Conference on Computer Vision*. Springer, 2010, pp. 368–381.

[4] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[5] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint," in *(CVPR),IEEE conference on*. Ieee, 2012.

[6] T. Trzcinski, V. Lepetit, and P. Fua, "Thick boundaries in binary space and their influence on nearest-neighbor search," *Pattern Recognition Letters*, vol. 33, no. 16, pp. 2173–2180, 2012.

[7] I. Jolliffe, *Principal component analysis*. Wiley Online Library, 2002.

[8] J. Philbin, M. Isard, J. Sivic, and A. Zisserman, "Descriptor learning for efficient retrieval," in *ECCV*. Springer, 2010.

[9] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *(CVPR'06)*. IEEE, 2006.

[10] D. Stavens and S. Thrun, "Unsupervised learning of invariant features using video," in *IEEE Conf. on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 1649–1656.

[11] S. Middelberg, T. Sattler, O. Untzelmann, and L. Kobbelt, "Scalable 6-dof localization on mobile devices," in *ECCV*. Springer, 2014, pp. 268–283.

[12] S. Lynen, M. Bosse, P. Furgale, and R. Siegwart, "Placeless place-recognition," in *2014 2nd International Conference on 3D Vision*, vol. 1. IEEE, 2014, pp. 303–310.

[13] S. Lynen, T. Sattler, M. Bosse, J. Hesch, M. Pollefeys, and R. Siegwart, "Get out of my lab: Large-scale, real-time visual-inertial localization," in *Robotics: Science and Systems*, 2015.

[14] B. Zeisl, T. Sattler, and M. Pollefeys, "Camera pose voting for large-scale image-based localization," in *Proceedings of the IEEE ICCV*, 2015.

[15] B. Schölkopf, A. Smola, and K.-R. Müller, "Kernel principal component analysis," in *International Conference on Artificial Neural Networks*. Springer Berlin Heidelberg, 1997, pp. 583–588.

[16] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[17] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[18] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[19] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *Journal of Machine Learning Research*, vol. 10, no. Feb, pp. 207–244, 2009.

[20] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg, "Matchnet: Unifying feature and metric learning for patch-based matching," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2015, pp. 3279–3286.

[21] K. Simonyan, A. Vedaldi, and A. Zisserman, "Learning local feature descriptors using convex optimisation," *IEEE PAMI*, vol. 36, no. 8, pp. 1573–1585, 2014.

[22] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "Lift: Learned invariant feature transform," in *European Conference on Computer Vision*. Springer, 2016, pp. 467–483.

[23] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer, "Discriminative learning of deep convolutional feature point descriptors," in *(ICCV)*, 2015.

[24] V. Balntas, E. Johns, L. Tang, and K. Mikolajczyk, "Pn-net: Conjoined triple deep network for learning local image descriptors," *arXiv preprint arXiv:1601.05030*, 2016.

[25] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.

[26] M. Schultz and T. Joachims, "Learning a distance metric from relative comparisons," 2004.

[27] Q. Ke and T. Kanade, "Robust l/sub 1/norm factorization in the presence of outliers and missing data by alternative convex programming," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 739–746.

[28] P. M. Long and R. A. Servedio, "Random classification noise defeats all convex potential boosters," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 608–615.

[29] T. Li, S. Zhu, and M. Ogihara, "Using discriminant analysis for multi-class classification: an experimental investigation," *Knowledge and information systems*, vol. 10, no. 4, pp. 453–472, 2006.

[30] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *IEEE CVPR*. IEEE, 2007, pp. 1–8.

[31] H. Jegou, F. Perronnin, M. Douze, J. Sánchez, P. Perez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE PAMI*, 2012.

[32] H. Jegou, F. Perronnin, M. Douze, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[34] A. Babenko and V. Lempitsky, "Aggregating local deep features for image retrieval," in *IEEE ICCV*, 2015, pp. 1269–1277.

[35] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *IEEE CVPR Workshops*, 2014.

[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[37] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE CCVPR*, 2015, pp. 1–9.

[38] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014.

[39] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[40] Y. Li, N. Snavely, and D. P. Huttenlocher, "Location recognition using prioritized feature matching," in *European conference on computer vision*. Springer, 2010, pp. 791–804.