# Visual localization by linear combination of image descriptors

Akihiko Torii, Josef Sivic, Tomas Pajdla

# Visual localization by linear combination of image descriptors

Akihiko Torii
Tokyo Institute of Technology
torii@ctrl.titech.ac.jp

Josef Sivic
INRIA*
Josef.Sivic@ens.fr

Tomas Pajdla
CMP, CTU in Prague
pajdla@cmp.felk.cvut.cz

## Abstract

*We seek to predict the GPS location of a query image given a database of images localized on a map with known GPS locations. The contributions of this work are three-fold: (1) we formulate the image-based localization problem as a regression on an image graph with images as nodes and edges connecting close-by images; (2) we design a novel image matching procedure, which computes similarity between the query and* pairs *of database images using edges of the graph and considering* linear combinations *of their feature vectors. This improves generalization to unseen viewpoints and illumination conditions, while reducing the database size; (3) we demonstrate that the query location can be predicted by interpolating locations of matched images in the graph without the costly estimation of multi-view geometry. We demonstrate benefits of the proposed image matching scheme on the standard Oxford building benchmark, and show localization results on a database of 8,999 panoramic Google Street View images of Pittsburgh.*

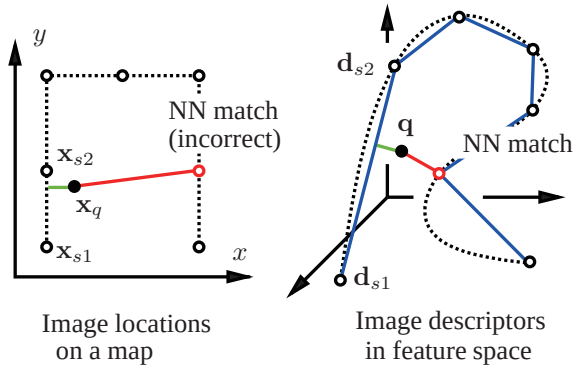


Figure 1. **Illustration of matching linear combinations of surrounding views on an image graph.** Left: Images localized along a path on a 2D map. Images are connected by edges (dotted lines), defining an image graph. The query image is shown in black. Right: Corresponding image descriptors in the feature space. Considering distances between the query descriptor and 1D subspaces (shown in blue) given by affine linear combinations of image descriptors along edges in the graph can lead to better matches. The nearest neighbor descriptor to the query is incorrect (red).

## 1. Introduction

The goal of this work is to predict the GPS location of a query image given a database of images with known GPS locations [29, 36]. This is a challenging task as the query and database images maybe taken from different viewpoints, under different illumination and partially occluded.

Significant progress has been recently achieved in large-scale localization using efficient representations from image retrieval [6, 15, 26, 28] often coupled with geometric constraints provided by 3D models of the environment [1, 11, 18].

We investigate a regression approach to the image-based location prediction problem and wish to find a mapping from some features of the query image to its position on the map, given a large database of geotagged images. The choice of the form of such a regressor is an important one and may depend on a number of factors including the feature representation, measure of image similarity, structure of the map and the number of images in the database.

We formulate visual localization as two stage regression: In the first stage, we wish to find a subset, which we call the *surrounding image set*, of images in the database, which depict the same place (i.e. the same 3D structure) as the query. In the second stage, the goal is to interpolate the location of the query from GPS locations of images in the matched subset of the database.

In the first stage we opt for the efficient bag-of-feature representation that has demonstrated excellent performance in large-scale image/object retrieval [5, 14, 22, 23] and place recognition [6, 15, 28]. In addition, we model the geotagged image database as an *image graph* [17, 25, 33, 37], where images are nodes and edges connect images at close-by locations on the map.

We design a matching procedure, illustrated in Fig. 1, that considers linear combinations of bag-of-feature vectors of database images along edges of the image graph.

*WILLOW project, Laboratoire d'Informatique de l'École Normale Supérieure, ENS/INRIA/CNRS UMR 8548.

Considering linear combinations of multiple feature vectors on the image graph can significantly improve matching accuracy by increasing robustness to missing or mismatched local image features, and improving generalization across viewpoint and illumination with only negligible effects on the computational cost at query time.

Given a set of matched images depicting the same 3D structure as the query, the camera of the query image can be estimated using structure from motion (SfM) techniques [8, 32]. SfM methods are getting mature, but they are computationally expensive and often depend on good initialization. Therefore, for the second stage, we investigate how well the location of the query can be predicted by interpolating GPS locations of the matched database images using image-based similarity alone without estimating the multi-view geometry of the underlying scene.

The contributions of this work are three-fold: (1) we formulate the image-based localization problem as a regression on an image graph; (2) we design a novel image matching procedure, which considers linear combinations of image feature vectors along edges in the graph; (3) we demonstrate that query image location can be directly predicted by interpolating locations of matched images.

**Related work:** Cummins and Newman [6], and Knopp *et al*. [15] perform large-scale appearance-based localization using the bag-of-feature representation but consider only matches to individual images in the database without considering linear combinations of bag-of-feature vectors. Schindler et al. [28] demonstrate appearance-based localization in a dense image sequence using vocabulary trees [22], spreading votes of individual query image features across close-by views on the map [19, 30, 35]. Zamir and Shah [35] localize query image using Google Street View imagery by matching individual feature descriptors (not quantized into visual words) and analyzing spatial distribution of the matches on the 2D map.

Our similarity measure considers linear combinations of feature descriptors is in spirit similar to query expansion [4, 5], which significantly improves object retrieval performance. We consider all linear combinations (rather than the average), expand the database (rather than the query) and compute the expansion in a closed form at query time, without ever computing and storing additional bag-of-feature vectors [5] or descriptors [24, 33] on the database side.

Irschara et al. [11] considered localization using SfM point clouds and generated synthetic views from a 3D model enabling generalization to unseen viewpoints. 3D point cloud is also used by Li et al. [18] who select a subset of 3D features that appear and were successfully matched in many database images. In contrast, we investigate localization by interpolating positions of database images without the costly 3D reconstruction, which might not be always available. We focus on street-side imagery such as Google Street View rather than video sequences [11] or landmark image collections [18].

Combinations of feature vectors lead to smaller image databases. Hence, our approach is also related to methods compressing the image datasets for localization and retrieval. Previous work in this area include methods based on image graphs [17, 25, 33, 37], epitomes [21], 3D point clouds [11, 18], or selecting representative features [15, 16, 28]. In particular, we build on the image graph methods, but create the graph using the structure of the 2D map (rather than matching images) and enhance generalization by considering linear combinations of feature vectors along edges of the graph.

Place recognition has been in part formulated as a linear classification task in a discrete set of predefined landmarks [17] or nearest-neighbor scene matching to obtain a coarse localization on the level of cities or continents [9]. In this work, we formulate the image-based localization as a regression problem and introduce a new, two-stage regression algorithm. Our regression approach is different from the standard regressors, such as e.g. kernel ridge regression [3], that typically consider only datapoint similarities in the feature space. In our case, we take advantage of the fact that images are organized on the map and build this structure into the regressor directly by considering linear combinations of pairs of spatially close images, given by the image graph.

## 2. Localization as regression on image graph

Next, we formulate the image-based localization task as a regression problem and outline the two-stage structure of the proposed regression function.

**The problem formulation:** We represent the visual content of images by their (tf-idf weighted [27]) bag-of-features [31] descriptors $\mathbf{d}$, which are vectors of dimension between $10^4$ and $10^6$, depending on the vocabulary size.

We consider an *image map* organized as a graph $G = (D, E)$ with $N$ vertices $D = \{(\mathbf{d}_i, \mathbf{x}_i)\}_{i=1}^{N}$ consisting of image descriptors and their corresponding locations on a planar map represented by two-dimensional vectors $\mathbf{x}_i$. Edges $E$ of $G$ link close views, which share visual content. In Google Street View data, for instance, $D$ corresponds to panoramic images and $E$ corresponds to links allowing to navigate from an image to its neighbours. In landmark datasets, such as the Oxford building benchmark set [23], edges may link images which share a planar structure which was successfully matched by a homography.

We introduce a localization function $f$

$$\mathbf{x}_q = f(\mathbf{q}, G) \tag{1}$$

that provides the location $\mathbf{x}_q$ of a descriptor $\mathbf{q}$ using an image map $G$.

We will demonstrate that edges $E$ provide a very convenient structure for image based localization and allow to localize query more accurately than individual image descriptors. Consider a pair of descriptors $\mathbf{d}_1$, $\mathbf{d}_2$, which are linked by an edge, i.e. close-by views in the image map. Now, imagine a query descriptor $\mathbf{q}$, from somewhere between the map views. Such $\mathbf{q}$ will likely share content with both descriptors $\mathbf{d}_1$, $\mathbf{d}_2$. It will be more similar to some combination of $\mathbf{d}_1$, $\mathbf{d}_2$ than to each of them individually. We will show that linear combination of $\mathbf{d}_1$, $\mathbf{d}_2$ is a meaningful combination which improves the performance of visual localization.

We will next focus on developing a suitable localization function $f$, given a visual map $G$ and a representative query set $D_Q$. In the experimental results, section 5, we will investigate localization performance with respect to different natural choices of $G$, corresponding to different densities of images in the image map. We focus on the analysis of the maximal error, and study the localization performance with respect to the percentage of query images localized within a given maximum distance.

**Two stage regression on an image graph:** We consider a set of localization functions constrained by the fact that we wish to predict the *relative* location of the query image w.r.t. only its *surrounding images* on the map, which share visual content with the query. The number of such surrounding images depends on the visible 3D structure (e.g. a narrow street vs. an open square), but in typical urban environments would be rather small.

Motivated by this constraint, it is natural to construct $f$ in two consecutive steps. In the first step, (a subset of ) the surrounding images that share the visual content with the query is identified and, in the second step, the location of the query w.r.t. the surrounding images is computed. This two-step approach is also desirable since efficient image search techniques [12, 22, 23] can be used to generate potential surrounding images and more expensive techniques of image matching with geometric verification can then be used within the surrounding set [23]. In this work, though, we will consider purely image-based approaches with no explicit use of multi-view geometry for the second step.

In particular, we consider a *linear regression* localization function $f$ of the form

$$\mathbf{x}_q = \sum_{s \in S(\mathbf{q})} w_s \mathbf{x}_s \tag{2}$$

where $\mathbf{x}_q$ is the predicted location of the query and

$$S(\mathbf{q}) = f_S(\mathbf{q}, G) \tag{3}$$

is the subset of $G$ determining the surrounding set for query vector $\mathbf{q}$ obtained by a *surrounding set retrieval* function $f_S$.

Next section describes details of obtaining the surrounding set for query vector $\mathbf{q}$ and section 4 describes how we obtain the predictive weights $w_s$, for a particular surrounding set.

## 3. Matching linear combinations of surrounding views

In this section we define surrounding image sets for a given visual map and explain how we perform retrieval by matching linear combinations of surrounding views linked by edges of $E$.

**Surrounding views.** Given query $\mathbf{q}$, we wish to determine graph $S(\mathbf{q}) = (D_S, E_S)$ – a subset of visual map $G$ – with images $D_S$ that share visual content with $\mathbf{q}$.

We observed that the query descriptor vector $\mathbf{q}$ is often approximated well by a linear combination of descriptor vectors of images linked by edges in $S(\mathbf{q})$. The intuition behind this approach is twofold. First, visual words present in the query vector, but missing in an individual database image, e.g. due to occlusion but also noise in detection and quantization, could be "filled-in" when multiple images are considered. Second, considering linear combinations of feature vectors is a form of view interpolation, but here performed in the high-dimensional feature space, which effectively expands the database to intermediate views.

With the above motivation, we are looking for a surrounding graph $S(\mathbf{q})$, which consists of only one edge from $G$. For instance, we exploit the structure of the Google Street View image database which is arranged into a planar graph such that images with small graph distance [7] share visual content. By restricting possible $S(\mathbf{q})$ to *pairs of images with unit graph distance*, we assure that images of $S(\mathbf{q})$ share visual content and can be meaningfully combined by the linear regressor (2). This choice is in particular appropriate for city street localization where only spatially close images, often captured in a sequence along a street, share useful visual content.

It will become clear from the experiments presented in section 5 that the nearest neighbor (NN) matching is sensible, especially for dense maps. For sparser maps, however, it is more successful to retrieve pairs of images with small graph distance. The intuition behind this observation is illustrated in Fig. 1.

The best matching pair $S(\mathbf{q})$ to the query $\mathbf{q}$ is obtained by finding the most similar affine combination among all pairs defined by $E$,

$$S(\mathbf{q}) = \text{argmin}_{(i,j)\in E} \min_{\alpha \in \mathbb{R}} \|\mathbf{q} - ((1-\alpha)\,\mathbf{d}_i + \alpha\,\mathbf{d}_j)\|^2, \tag{4}$$

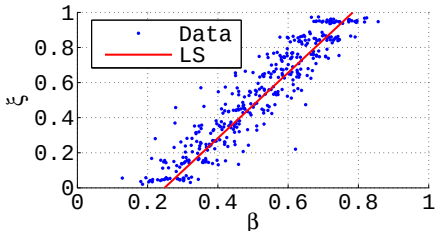


Figure 2. Values of the spatial interpolator $\xi$ as a function of the relative similarity $\beta$, given by (6), for many training query images sampled along lines connecting image map images. Notice the affine dependence of $\xi$ on $\beta$.

where $\alpha$ is the parameter of the affine combination. The minimum over $\alpha$ can be obtained in a closed form as

$$\alpha^* = \frac{(\mathbf{d}_j - \mathbf{d}_i)^\top (\mathbf{q} - \mathbf{d}_i)}{\|\mathbf{d}_j - \mathbf{d}_i\|^2}. \quad (5)$$

Linear and convex combinations can be also considered. In our case, however, we found that there was no difference between the affine and convex combination and only a very small (a fraction of a percent) difference in performance between the affine and linear combination. The advantage of the affine (and convex) combination is that it is determined by one parameter and hence it facilitates one-dimensional linear regression of the next prediction step (Section 4).

Note also, that as the minimum over $\alpha$ can be computed in a closed form, considering linear combinations of feature vectors has only a small effect on the efficiency of the localization. In particular, as in object retrieval [23, 31] efficient inverted file indexing [34] can be used to compute inner products between the query and database vectors, which are in turn used to compute optimal $\alpha$ for each pair. In addition, for typical image maps the number of considered image pairs (i.e. edges in the image graph) grows only linearly with the number of images in the image map.

Equipped with the surrounding set $S(\mathbf{q})$ for the query, composed of a pair of images, we now consider the prediction of the query location.

## 4. Predicting position of the query image

In this section, we discuss how to obtain weights $w_s$ in the predictor given in (2). While it may seem that the affine combination parameter $\alpha^*$ can be used directly for predicting the location of the query, it turns out that such predictions are significantly biased towards the mean of locations in the pair of surrounding views. This might be explained by missing visual words in the query with respect to its surrounding views, due to occlusions, missed detections or variations beyond invariance built into the descriptor vector. To see this, consider an example where the query image is taken at exactly the same position as the first image in the surrounding set, i.e. $\mathbf{x}_q = \mathbf{x}_1$. Ideally, from (2), the desired output would be $w_1 = 1$ and $w_2 = 0$. However, due to potential differences in lighting, occlusions and measurement noise, $\mathbf{q} \neq \mathbf{d}_1$ and hence $\alpha \neq 0$.

We found that the best predictor of query location is obtained as follows. First, we compute the relative similarity of query to both images in the best matching surrounding set,

$$\beta = \frac{\mathbf{q}^\top \mathbf{d}_{s1}}{\mathbf{q}^\top \mathbf{d}_{s1} + \mathbf{q}^\top \mathbf{d}_{s2}}. \quad (6)$$

We examine many descriptors $\mathbf{q}$ and their corresponding surrounding images $S(\mathbf{q}) = (s_1, s_2)$ and express the position of $\mathbf{q}$ as the affine combination of its surrounding images $\mathbf{x}_q = \mathbf{x}_{s_1} + \xi\,(\mathbf{x}_{s_2} - \mathbf{x}_{s_1})$. We thus obtain the corresponding location interpolator $\xi$. Then, we have also obtained the corresponding relative visual similarity $\beta$ according to (6).

Fig. 2 shows values of $\xi$ against values of $\beta$ for many real images. We can see that there is a strong affine dependence of $\xi$ on $\beta$ and therefore it is possible to estimate $\xi$ from $\beta$ by an affine interpolator

$$\xi_q = a_0 + a_1 \beta. \quad (7)$$

Given this observation, we train parameters $a_0, a_1$ on an independent representative set of images and then use them to predict query position $\mathbf{x}_q$ as

$$\mathbf{x} = \mathbf{x}_{s_1} + (a_0 + a_1\beta)(\mathbf{x}_{s_2} - \mathbf{x}_{s_1}) \text{ with } (s_1, s_2) = S(\mathbf{q}) \quad (8)$$

where $S(\mathbf{q})$ is determined by (4) and $\beta$ by (6).

## 5. Results

We evaluate the benefits of considering pairs of feature descriptors along edges in the image graph for image matching and location prediction.

**Matching performance.** The matching performance is evaluated on two tasks: (i) matching panoramic images from Google Street View and (ii) matching perspective images from the standard Oxford Buildings benchmark [23].

In the first task, we use the Google Street View Pittsburgh Research Data Set ("Street View Research"), which consists of a sequence of 8,999 panoramic images. We first downsample all images to 1,664 $\times$ 512 pixels (half the original size) to reduce the number of feature detections. The images are then described by the bag-of-features representation [31, 23]: 1) SURF descriptors [2] are extracted, 2) visual vocabulary of 100,000 visual words from SURF's from every 20th image is built by approximate k-means clustering [20, 23], 3) tf-idf weighted vectors [27] are computed by quantizing the SURF descriptors using the visual vocabulary and re-weighted according to the number of occurrences of each word in the database, 4) tf-idf vectors are normalized to unit length and 5) the image similarity is computed as the inner product of tf-idf vectors.

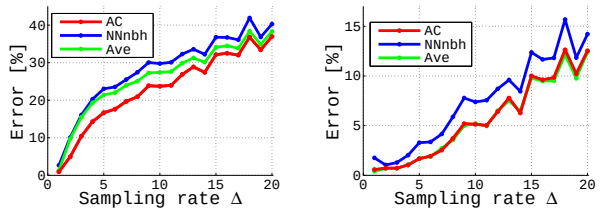


Figure 3. Matching performance for the proposed method (AC) (red), the similarity averaging approach (Ave) (green) and the baseline nearest neighbor based approach (NNnbh) (blue). The error is measured as the percentage of mismatched query images (y-axis) for different database sampling rates $\Delta$ (x-axis). A mismatch is declared when the query is not assigned (a) to the correct (ground truth) edge in the graph or (b) to the correct edge or its neighbors in the graph (a less strict criteria than (a)). Note the significant reduction in error rate for AC compared to baseline NNnbh. AC outperforms the Ave method if exact match is required (a), but both methods perform similarly in coarse-level matching (b).

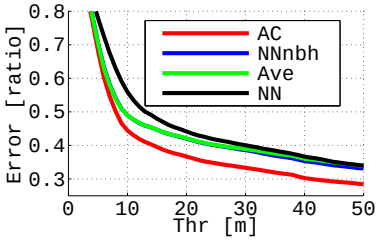


Figure 4. **Matching pairs of Street View images.** Y-axis shows the fraction of incorrectly matched query test images ("Street View Research"). Query image is considered mismatched if the best matched database image ("Street View Web") is further than "Thr" meters (x-axis) from the ground truth nearest neighbor in the database. Note the reduction in matching error by the proposed linear combination matching method (AC) compared to the three other baselines (see text for details).

We use subsets of the odd numbered frames as the image database $D$ with known GPS locations and all even numbered frames as the query set $D_Q$. The image map, represented by graph $G$, is constructed by regularly sampling every $\Delta$-th image from the database $D$. We call $\Delta$ *sampling rate*. Edges of the graph are obtained by connecting consecutive images along the (known) acquisition sequence. In the Street View Research Set, where images are roughly spaced by 1 meter, sampling rate of $\Delta \in \{5, 10, 15, 20, 25\}$ corresponds roughly to image spacing of $\{10, 20, 30, 40, 50\}$ meters (recall that half the original images are set aside as the query set). Note that by construction, each query image from the query set $D_Q$ lies along one edge of graph $G$, which defines the ground truth surrounding set for the query. The goal of matching is to find the correct surrounding set for each query. We investigate matching performance of the linear combination method (AC) and the averaging method (Ave), which finds the best matching pair by averaging the image descriptor vectors along the edge (equivalent to fixing $\alpha = 0.5$, or averaging the similarity scores). The results are compared to a baseline method (NNnbh), which finds the best nearest neighbor image followed by finding the second nearest neighbor within the two surrounding locations of the best match. Fig. 3 shows the matching performance with respect to varying database sampling rate $\Delta$.

In the next experiment we use 1,143 Pittsburgh Google Street View images ("Street View Web") as the image map. The image graph is defined by the connectivity between the images as used to navigate on maps.google.com. Images and the graph structure were downloaded from maps.google.com. All images from the Street View Research dataset are used as queries. This is a realistic and extremely challenging scenario, where (most of) the database and test images are obtained under different acquisition conditions (different season, time of day, path, etc). The spacing between the database Street View Web images is between 10-20 meters, which corresponds approximately to sampling rates of 5-10 on the Street View Research data. The proposed approach (AC) is compared to the nearest neighbor matching baseline (NN), its modification (NNnbh) and the averaging based method (Ave) described above. Results are shown in Fig. 4 and examples of correctly and incorrectly matched queries are shown in Fig. 5.

We also evaluate image pair matching on the Oxford Building Dataset [23], a standard benchmark dataset for object retrieval. We use the tf-idf vectors from [10] and obtain the image graph from the authors of [25]. The image graph has 34,028 edges and is built based on the efficient pairwise matching with geometric verification using planar homographies to ensure that images connected by edges contain the same 3D structures. Retrieval performance is evaluated using the standard mean average precision (mAP) over all 55 queries. Given a query, retrieval using the image graph is performed as follows: (i) all nodes are scored using the similarity (inner product) to the nearest neighbor (NN) [23]; (ii) all edges are scored using the proposed linear combination method; (iii) images are ranked using the highest scoring edge (if connected) or their individual score (if singletons). We compare four different edge scoring methods for retrieval using the image graph: (a) the linear combination approach and (b-d) the average/max/min similarity of the two images along the edge. Results are compared to the nearest neighbor (NN) [23] and NN with soft-assignment (SA) [24] baselines in table 1. Example matches are shown in Fig. 6. All image graph methods significantly improve over the baseline nearest neighbor (NN). In addition, combining images along edges in the graph, either by considering their linear combinations (AC) or averaging their sim-

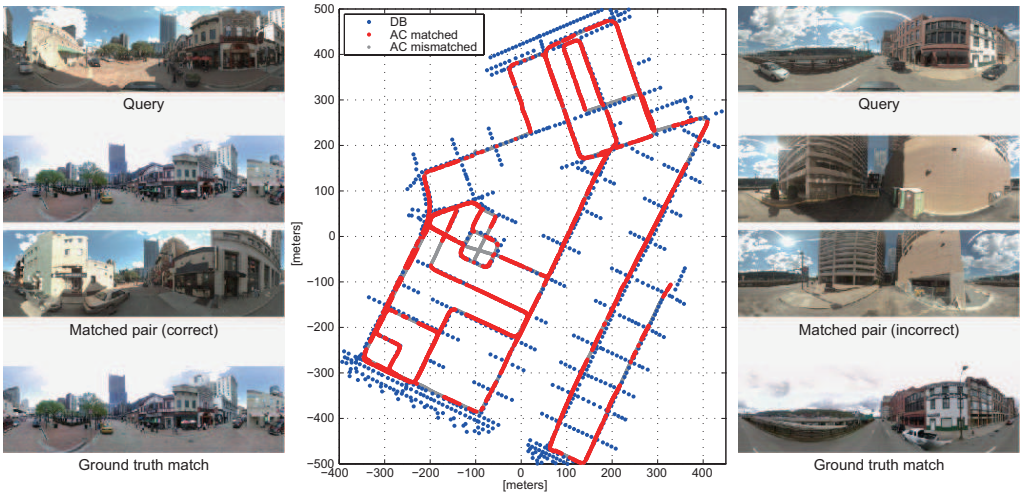


Figure 5. Examples of a matched (left column) and a mismatched (right column) query. The map (middle column) shows all matched (red) and mismatched (gray) query images ("Street View Research"). Database images ("Street View Web") are shown in blue. Query image is considered mismatched if the best matched database image is further than 20 meters from the location of the ground truth nearest neighbor.

| Baselines | | Image graph methods | | | |
|---|---|---|---|---|---|
| NN [23] | SA [24] | AC | Ave | Max | Min |
| 0.6138 | 0.6694 | **0.7589** | 0.7546 | 0.7020 | 0.6138 |

Table 1. Mean average precision (mAP) on the Oxford building dataset for the baseline nearest neighbor (NN) and nearest neighbor with soft-assignmnet (SA) methods compared with different image graph retrieval methods that combine images along edges in the graph.

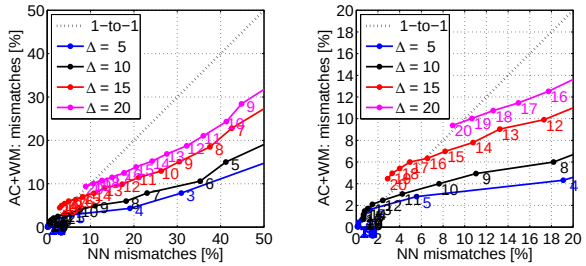


Figure 7. Localization error measured by the percentage of mismatches of the proposed method (AC) (y-axis) vs. the baseline nearest neighbor approach (NN) (x-axis). Each curve shows accuracy for a fixed sampling rate $\Delta$. Each marked point on the curve corresponds to a particular value of the localization tolerance $gt_{dist}$ in meters. The values of $gt_{dist}$ are shown next to each point. (b) is a close-up of (a) at the origin.

ilarities to the query (Ave), improves over considering images along the edge individually (Max / Min methods) as well as the soft-assignment (SA) baseline. Although the AC method performs the best, the Ave method performs also well. This maybe due to the nature of the benchmark, which evaluates retrieval (i.e. finding *all* matches containing the query object) as opposed to localization (accurately finding the closest matching pair), where estimating the precise value of $\alpha$ may not be that important. To further evaluate benefits of linear combination matching a dataset with accurate image locations will be necessary. The proposed graph-based matching/localization approaches (AC or Ave) are also memory efficient, as no new views need to be synthesized [11] or individual descriptors propagated to neighboring views [33]. Finally, further matching improvements are expected by combining the proposed approach with the expansion on the query side [4, 5].

**Predicting query location.** Here we evaluate the localization accuracy of the complete two stage regressor using the Street View Research dataset.

The data is split again into the database set $D$ and the query set $D_Q$ as outlined above. The last 900 frames of the sequence are used as training data to obtain the regression parameters in (7) and are removed completely from the query set and the database. We measure the localization performance by the percentage of query images, which are localized within $gt_{dist}$ meters from their ground truth GPS location. Again, we investigate performance when varying this localization accuracy threshold.

We compare performance of the proposed method (AC) with the standard nearest neighbor baseline (NN), where the query location is predicted as the location of the nearest neighbor to the query. We have also experimented with variations of the NN approach where, the query location is predicted as a (learnt) linear combination of the two nearest neighbor locations in the feature space or a (learnt) linear combination of the nearest neighbor and the second nearest neighbor within surrounding locations. However, we found

Figure 6. The benefits of linear combination matching for retrieval on the Oxford building dataset. Each triplet shows: query image (left) and a pair of images along an edge in the image graph, which were low ranked individually, but matching the query to the linear combination of their tf-idf vectors significantly improved their ranking.

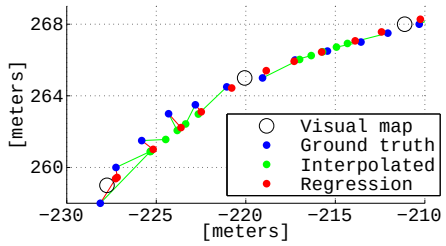


Figure 8. Localization with and without the learnt regressor (best viewed in color). Black circles "o" denote known positions of images in the image map. Blue dots denote the ground truth locations of the query images. Predictions obtained by interpolation using the relative similarity score $\beta$ directly are shown in green. Predictions obtained by the corrected weights of the combined two-stage regressor are shown in red. Both types of predictions are connected by lines to the corresponding ground truth locations. Note how the regressed predictions (red) improve localization over the directly interpolated predictions (green), which tend to cluster in the middle of the matched image pairs in the image map.

that the simple NN approach performs the best.

Fig. 7 compares the localization performance between the proposed method (AC) (y-axis) and the baseline nearest neighbor approach (NN) (x-axis). When the curves lie under the dashed line the proposed method AC performs better than baseline NN. It is interesting to study the result of sampling rate $\Delta = 5$ and 10, which corresponds to blue and black curves, because the spacing of 10 to 20 meters is roughly the spacing of Google Street View images available on the Internet. The improvement by AC is significant especially with small localization tolerance threshold $gt_{dist}$. For example, for $\Delta = 5$ (see the blue curve in Fig. 7(b)), which corresponds roughly to spacing of 10 meters between images in the image map, and localization tolerance $gt_{dist} = 5$ meters, the nearest neighbor approach incorrectly localizes about 6% of query images, while the proposed approach reduces this error to half (about 3%). Fig. 9 shows examples of correctly localized queries using the proposed linear combination matching. Fig. 8 illustrates the benefits of using the learnt two-stage regressor over using directly the relative similarity score $\beta$ to predict the query location.

Note that the better performance of the proposed approach over the baseline NN method is due to two reasons. First, the proposed method can find the correct surrounding set, in cases when the NN method fails, i.e. considering linear combinations of close-by views provides better matches to the query. Second, once the correct surrounding set is found, the proposed method can more accurately predict a the location of the query as a (learnt) linear combination of locations of the images in the surrounding set.

**Scalability:** Practical image based localization has to aim at working with millions of images. The state of the art approaches to image search [12, 22, 23] are based on efficient search for the single most similar (NN) descriptor.

In this work we replace the search for the single most similar descriptor by a search for a pair of descriptors. Although searching for pairs might be too expensive in general, the structure of visual localization makes searching for pairs (almost) as efficient as searching for individual descriptors.

Consider that each image in the image map has only a small number of its potential surrounding sets and therefore the size of surrounding sets is linear (with the slope 1 to 3) in the number of images in the database. For instance, if the database of $N$ images is arranged as a sequence, there is only $N - 1$ surrounding pairs to consider. In case of fully planar localization, when image pairs would be replaced by image triplets, the number of triangles to consider would be maximally $3\,N - 6$ (consider Euler formula for a planar triangulated graph). This makes our technique as scalable as the search for the most similar descriptor at a considerably better performance.

## 6. Conclusions

We have formulated the image-based localization task as a regression problem on an image graph and developed a two stage regressor, which takes advantage of the graph structure of the database. We have shown that considering linear combinations of descriptors along edges in the image graph significantly improves matching accuracy over the standard nearest neighbor matching. We have considered linear combinations of two close-by images, which is well suited for matching street-side imagery. The concept, however, can be generalized to image triplets or n-tuples, which

(a) Query image 6888 (b) i1 6881 (c) i2 6891 (d) 1st NN 371 (e) 2nd NN 6881

(a) Query image 2628 (b) i1 2621 (c) i2 2631 (d) 1st NN 1841 (e) 2nd NN 2631

Figure 9. **Examples of correct localization on the map with sampling rate $\Delta = 5$ using the proposed method.** (a) Query image. (b-c) The image pair selected by the proposed matching of linear combinations of image descriptors (correct). (d-e) The first two nearest neighbors (1st NN is incorrect). The distance between images (b) and (c) on the visual map is about 10 meters. The predicted location of the query is within 3 meters of its true location.

may be considered for other scenarios where the 2D visual map is densely connected. Finally, we have used the bag-of-features representation, but the proposed approach can be used with other image descriptors developed for large-scale matching [13].

## Acknowledgements

This work was partly supported by the MSR-INRIA laboratory, the ANR project DETECT (ANR-09-JCJC-0027-01), the EIT ICT labs (activity 10863), ME CR MSM 6840770038, CTU SGS10/190/OHK3/2T/13 and FP7-SPACE-241523 PROVISCOUT grants.

## References

[1] G. Baatz, K. Koser, G. R., and M. Pollefeys. Handling urban location recognition as a 2D homothetic problem. In *Proc. ECCV*, 2010.

[2] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *Proc. ECCV*, 2006.

[3] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[4] O. Chum, A. Mikulik, M. Perdoch, and J. Matas. Total recall II: Query expansion revisited. In *CVPR*, 2011.

[5] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *ICCV*, 2007.

[6] M. Cummins and P. Newman. Highly scalable appearance-only SLAM - FAB-MAP 2.0. In *Proceedings of Robotics: Science and Systems*, Seattle, USA, June 2009.

[7] R. Diestel. *Graph Theory*. Springer-Verlag, 2000.

[8] R. I. Hartley and F. Schaffalitzky. Reconstruction from projections using Grassmann tensors. In *Proc. ECCV*, 2004.

[9] J. Hays and A. Efros. im2gps: estimating geographic information from a single image. In *CVPR*, 2008.

[10] http://www.robots.ox.ac.uk/∼vgg/data/oxbuildings/.

[11] A. Irschara, C. Zach, J. Frahm, and H. Bischof. From structure-from-motion point clouds to fast location recognition. In *CVPR*, 2009.

[12] H. Jegou, M. Douze, and C. Schmid. On the burstiness of visual elements. In *CVPR*, 2009.

[13] H. Jegou, M. Douze, C. Schmid, and P. Perez. Aggregating local descriptors into a compact image representation. In *CVPR*, 2010.

[14] H. Jegou, H. Harzallah, and C. Schmid. A contextual dissimilarity measure for accurate and efficient image search. In *CVPR*, 2007.

[15] J. Knopp, J. Sivic, and T. Pajdla. Avoiding confusing features in place recognition. In *Proc. ECCV*, 2010.

[16] F. Li and J. Kosecka. Probabilistic location recognition using reduced feature set. In *Proc. Int. Conf. on Robotics and Automation*, 2006.

[17] Y. Li, D. Crandall, and D. Huttenlocher. Landmark classification in large-scale image collections. In *ICCV*, 2009.

[18] Y. Li, N. Snavely, and D. Huttenlocher. Location recognition using prioritized feature matching. In *Proc. ECCV*, 2010.

[19] D. Lowe. Local feature view clustering for 3D object recognition. In *CVPR*, pages 682–688. Springer, 2001.

[20] M. Muja and D. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *VISAPP*, 2009.

[21] K. Ni, A. Kannan, A. Criminisi, and J. Winn. Epitomic location recognition. *IEEE PAMI*, 2009.

[22] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *CVPR*, 2006.

[23] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007.

[24] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR*, 2008.

[25] J. Philbin, J. Sivic, and A. Zisserman. Geometric latent dirichlet allocation on a matching graph for large-scale image datasets. *IJCV*, 2010.

[26] A. Roshan and M. Shah. Accurate image localization based on google maps street view. In *Proc. ECCV*, 2010.

[27] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5), 1988.

[28] G. Schindler, M. Brown, and R. Szeliski. City-scale location recognition. In *CVPR*, 2007.

[29] S. Se, D. G. Lowe, and J. Little. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *Intl. J. of Robotics Research*, 21(8):735–758, 2002.

[30] C. Silpa-Anan and R. Hartley. Localization using an image-map. In *ACRA*, 2004.

[31] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, 2003.

[32] N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3D. In *SIGGRAPH*, 2006.

[33] P. Turcot and D. Lowe. Better matching with fewer features: The selection of useful features in large database recognition problem. In *WS-LAVD, ICCV*, 2009.

[34] I. H. Witten, A. Moffat, and T. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann Publishers, ISBN:1558605703, 1999.

[35] A. Zamir and M. Shah. Accurate image localization based on google maps street view. In *Proc. ECCV*, 2010.

[36] W. Zhang and J. Kosecka. Image based localization in urban environments. In *3DPVT*, 2006.

[37] Y. Zheng, M. Zhao, Y. Song, H. Adam, U. Buddemeier, A. Bissacco, F. Brucher, T. Chua, and H. Neven. Tour the world: building a web-scale landmark recognition engine. In *CVPR*, 2009.