Absolute geo-localization thanks to Hidden Markov Model and exemplar-based metric learning

Cédric Le Barz
THALES Services - 91767 Palaiseau, France
cedric.lebarz@thalesgroup.com

Nicolas Thome, Matthieu Cord
Sorbonne University, UPMC University, Paris 06, UMR 7606, LIP6 - 75005 Paris, France
nicolas.thome@lip6.fr, matthieu.cord@lip6.fr

Stéphane Herbin and Martial Sanfourche
French Aerospace Lab, ONERA - 91123 Palaiseau, France
stephane.herbin@onera.fr, martial.sanfourche@onera.fr

Abstract

This paper addresses the problem of absolute visual ego-localization of an autonomous vehicle equipped with a monocular camera that has to navigate in an urban environment. The proposed method is based on a combination of: 1) a Hidden Markov Model (HMM) exploiting the spatio-temporal coherency of acquired images and 2) learnt metrics dedicated to robust visual localization in complex scenes, such as streets. The HMM merges odometric measurements and visual similarities computed from specific (local) metrics learnt for each image of the database. To achieve this goal, we define some constraints so that the distance between a database image and a query image representing the same scene is smaller than the distance between this query image and other neighbor images of the database. Successful experiments, conducted using a freely available geo-referenced image database, reveal that the proposed method significantly improves results: the mean localization error is reduced from 12.9m to 3.9m over a 11km path.

1. Introduction

The problem tackled in this paper is the visual geo-localization of a vehicle operating in an urban environment. Visual ego-localization is a key function for autonomous vehicles such as personal service vehicles, self-driving cars, and unmanned aerial systems, as it allows these systems to navigate autonomously within their environment in order to perform their mission. Visual geo-localization or visual place recognition is a challenging task because two images of the same place acquired at different times and with different cameras may show huge appearance differences due to illumination and colorimetry variations (*e.g.* sunny

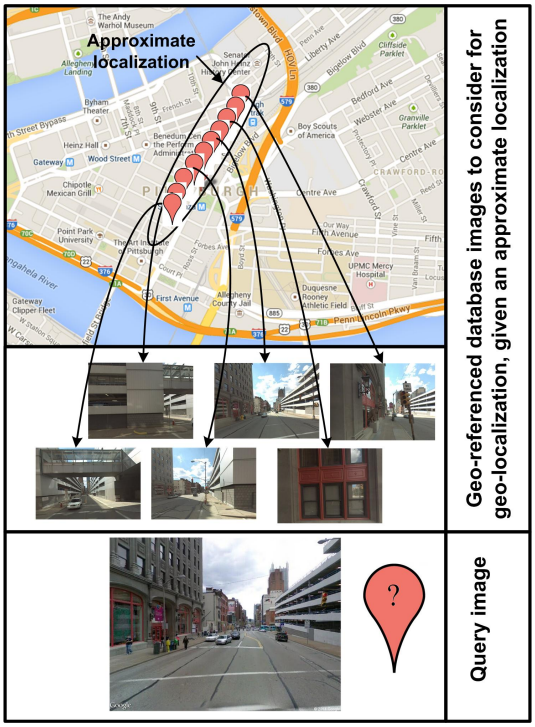
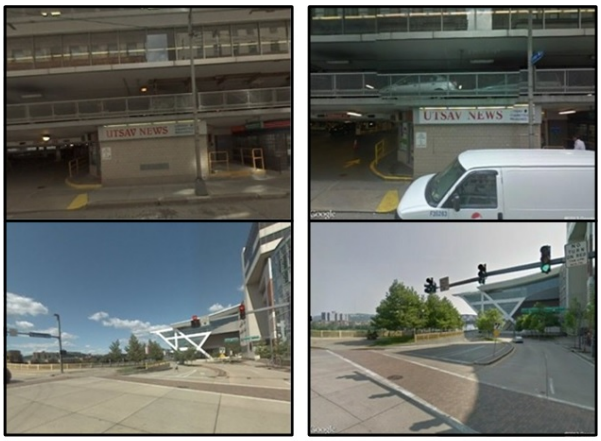


Figure 1. Our system aims at answering the following question: knowing a roughly position of the vehicle in a street and the scene being observed by the vehicle's camera, can we determine where is it exactly along the street?

or cloudy days), camera viewpoints changes, scene modifications (*e.g*. seasonal changes, building construction) and occlusions (*e.g*. by cars) (Fig. 2). Robustness to large variability in scene appearance is required for all autonomous systems aiming at long-term operations in both indoor and outdoor environments. It has to be noted that, even if it exists absolute localization systems like GPS, such systems are not robust and precise enough. Furthermore, even if odometric systems, IMU based or visually based, are able to provide relative localization information at low cost, their information is relative to a given position and suffers from drift especially on complex trajectories. It can only be used reliably on small portions of a trajectory and can't be the only source of measurement for absolute localization. We propose a visual geo-localization solution including a novel similarity estimator dedicated to applications for which geographical positions (GPS) of database images are available and an *a priori* approximate localization of query image is known. The solution aims at localizing precisely the vehicle so that it can follow the trajectory that has been previously defined (Fig. 1).



(a) (b)

Figure 2. Vehicle images (a) and Google Streetview images (b). Note the impact of different focal lenses, weather conditions, viewpoint changes and the presence/absence of cars in the scene.

2. Related work

Visual geo-localization methods are mainly based on image retrieval (IR) algorithm [26] [12] [14]. Most of them rely on the extraction of features, that are compared directly (kNN vote) or indirectly (Bag Of Words model) [27][13][1] to geo-referenced features database using a distance. Methods mainly differ on the type of features extracted online, the matching method, and the *a priori* information used (geo-referenced database image, 3D model, 2D roadmap,...). Three different targeted types of application can be distinguished, depending on the targeting localization precision, i.e. world scale localization ([12][8]), city scale localization ([26][11]) and street scale localization ([30][28][2] [3][22][19]).

Some examples of visual geo-localization methods are briefly described hereafter. Schindler *et al*. [26] presented a method for city scale localization based on the Bag Of Words signature (BOW) using a dataset of street side images. They proposed a greedy algorithm in order to improve the accuracy of image retrieval for large scale database image by optimizing vocabulary trees. Zamir *et al*. proposed in [30] a hierarchical method to localize a group of images. SIFT descriptors from database images are indexed using a tree. A nearest neighbor tree search is then computed for each SIFT query image feature. Weak votes are removed and each reliable feature votes for a location. All accumulated spatial votes are then filtered by a Gaussian kernel. The geo-referenced image with the highest number of votes determines the location. In [28], the method described in [30] is improved by interpreting the 2D map votes as a likelihood. This likelihood is then used in a Bayesian tracking filter to estimate the temporal evolution based on the previous state. Both solutions are dedicated to web video annotation, and localization is not realized on the fly. In [2], the vehicle localization algorithm uses simple visual features and 3D features. The solution requires that a compact map described as a graph is built in a preliminary phase. Nodes include vehicle position at fixed distance interval and visual and 3D features. At runtime, a Bayesian filter is used to estimate the probability of the vehicle position by matching the features extracted from sensors with database features. Their solution uses two lateral cameras and two lateral LIDARs. The same sensors are used during the map building step and the localization step. In contrast, our solution is monocular and uses different cameras for acquisition and reference database. In [22], the localization is achieved by recognizing temporal coherent sequences of local best matches. These local best matches are based on a Sum of Absolute Difference (SAD) on resolution-reduced and patch-normalized images between last acquire image and M previous images. The proposed solution is robust to extreme perceptual changes. In [23], the solution has been improved to provide invariance to vehicle speed, but remains sensitive to important point of view variations. Our solution adresses this problem by learning. In [19], authors work on visual similarity for UAV ego-localization. They propose to generate artificial views of the scene in order to overcome the large view-point differences. Nevertheless, spatio-temporal constraint is not taken into account. Loop closure algorithms include place recognition functionnality. One state-of-the-art example is FAB-MAP [6] [7]. Authors propose a probabilistic model on the top of bag of visual words to compute the probability that two observations are collected from the same location. The model takes into account the correlation

between visual words. Their solution remains sensitive to strong perceptual changes, and the same camera is used for database images and query images. Odometry information has been incorporated into FAB-MAP [18]. Another type of approach is to learn specificities of each place, as our solution that learns a visual similarity measure for each place. In [11], the problem is cast as a classification task, a classifier for each image in the database is trained using per-exemplar SVM approach. In [21] the authors propose to learn a bank of detectors for each place, to identify specific scene structures instead of local features.

Our solution is dedicated to street scale localization. As in [30] [28] [2] [22] [23], our solution uses spatio-temporal coherency thanks to the use of a Bayesian filter enabling to take into account visual similarities and odometrics measurements. No assumption is done concerning the constant velocity of the vehicle, but as in [31] we consider coarse position estimates provided by an odometric sensor and their uncertainties. As in [19], we generate artificial view from available geo-referenced database images. Our similarity visual solution, Exabal, uses these artificial views to learn offline, in a supervised way, a local similarity measure dedicated to robust visual localization in complex scenes such as street images. Exabal learns a local pseudo-distance matrix for each image of the database, so that the distance between a database image and the given query acquired from the same scene is smaller than the distance between this query and other (close) images in the database.

Our method encompasses the following contributions:

- We propose a novel local visual similarity measure included within a HMM. The learning of a local metric for each dataset image is cast as a convex optimization problem, which is efficiently solved with a projected gradient descent scheme. Once this off-line procedure is carried out, computing the similarity at test time is very efficient, *e.g*. much faster than methods based on descriptor votes like [30]. It can benefit from existing fast indexing structures (*e.g*. inverted files, search trees).

- During training, we generate sensible geometric and photometric transformations to model images similar to unknown query images. This makes possible the learning of features able to discriminate a given dataset image from its neighbors, and at the same time to learn invariance to common transformations occurring at test time.

- Successful experiments reveal the ability of the method to localize the vehicle within complex street scene. We show that the model learns sensible invariances and discriminating features for localization.

The effectiveness of our approach has been evaluated over a 11 km path using two kinds of images: Google Streetview images [4] simulating images acquired online by the vehicle camera and Google Pittsburgh image dataset [5] as geo-referenced image database.

3. Exabal-HMM solution

The learnt local visual similarities, provided by Exabal, are included within a Bayesian based solution, a HMM, which is described in details in section 3.1, while our similarity measures based on an exemplar-based metric learning solution is described in section 3.2. The HMM used is similar to the one described in [17]. It enables to filter out some wrong matches provided by standard image retrieval algorithms, by finding the trajectory that best explains the M past observations and, therefore, current vehicle position. Such solution is an hybrid framework allowing to take into account uncertainties of estimated position, odometric measurements, and visual similarity measurements.

3.1. Position tracking with HMM

3.1.1 General principle

At each time t the vehicle acquires an image O_k and receives an estimate of its current position $\tilde{S}_k$ from the odometric system. The goal of the absolute localization algorithm is to produce a better estimate $\hat{S}_k$ of the current vehicle position from the past observations and odometric measurements (Fig. 3). The estimator is a function of the M past observations $\mathbf{O}_k = \{O_{k-M+1}, \dots O_k\}$ (i.e. the current location estimate exploits a set of observations in a sliding window based approach of length M) and the estimated position $\tilde{S}_k$. Estimation is realized in a classical

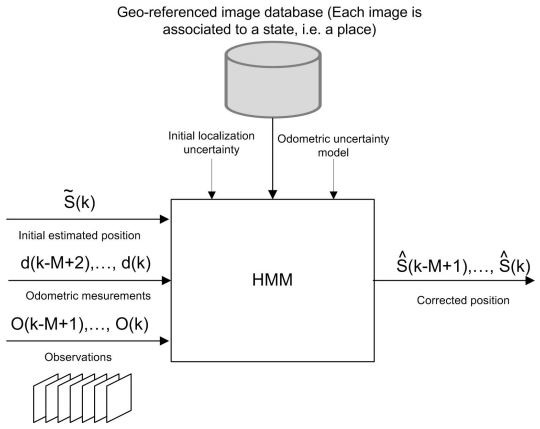


Figure 3. System overview: from the M past observations and associated odometrics measurements as well as a rough position estimate $\tilde{S}_k$ the system combines within a HMM all information to determine the corrected position $\hat{S}_k$.

random variable setting where the vehicle location at time t

is considered as a random variable q_t taking values in a discrete set of possible location S_j with $j \in \{1, ..., N\}$. The main modelling hypothesis is that its random behaviour is represented by a HMM.

Using the classical notations of [24], the use of a HMM requires the definition of the adequate model $\lambda = \{N, M, \Pi, A, B\}$ where N is the number of states, M is the number of observations, Π is the prior on the initial state (*i.e.* the estimated position), A is the transition probability matrix between the states and B is the observation probability matrix given some states.

The HMM approach provides a standard way to estimate the most likely state sequence $\hat{\mathbf{S}}_k$, i.e the M successive places, explaining the sequence of observations $\mathbf{O}_k = \{O_{k-M+1}, ..., O_{k-1}, O_k\}$:

$$\hat{\mathbf{S}}_k = \arg\max_{\mathbf{S}} P(\mathbf{S}|\mathbf{O}_k, \lambda) \quad (1)$$

This can be solved with the Viterbi algorithm.

The question is now to design the HMM adapted to the estimate of the absolute vehicle location. This will be detailed in two steps: construction of the state transition matrix A and initial state vector, and computation of the conditional observation matrix B.

3.1.2 State transition matrix and initial state vector

The state transition matrix A and initial state vector are built from knowledge of the odometric system behaviour and vehicle kinematics. From the vehicle kinematics, images are approximately acquired every D meters [1] with an odometric uncertainty of Δ meters. The image database consists of overlapping images acquired every $D^{'}$ meters with $D^{'} \leq D$. In this setting, the database is therefore assumed to have a bigger sampling rate than the online image acquisition rate. Each possible state location S_j is uniquely defined by a geo-referenced database image I_j.

The filtering capacity of the HMM depends on the number M of past observations. One critical parameter is the localization uncertainty U which defines the area where the vehicle is supposed to be. This localization uncertainty can be, for example, the initial position uncertainty when the vehicle starts its planned trajectory.

The number of states N, i.e. the number of potentially matching images in the database, the initial state probability Π and the state transition probability matrix A depend on U, D, Δ and M. They are defined the following way:

- N : Given the putative position of the vehicle $\tilde{S}_k$, the localization uncertainty U, the approximate displacement D, and the observation number M, the potential states, i.e. the set of database images to consider for matching can be easily determined.

[1] For simplicity, D is supposed to be constant, but it could be variable.

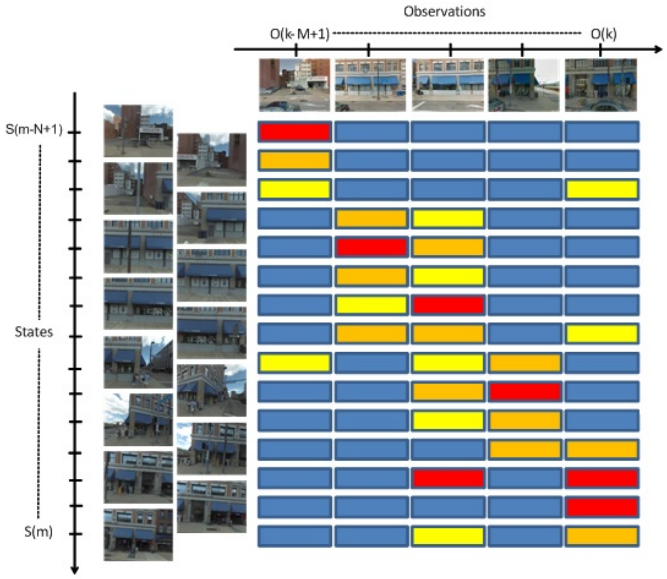


Figure 4. Observation matrix $\mathbf{B}$: the HMM enables to find the trajectory among these visual similarities taking into account uncertainties of initial estimated position ($\mathbf{\Pi}$ vector) and odometric measurements, i.e. possible transitions ($\mathbf{A}$ matrix). Red color means high similarity, while blue color means low similarity.

- Π: $\Pi = \{\pi_j\}_{j=1}^{j=N}$ where $\pi_j = P[q_1 = S_j]$. It depends on initial position estimate (i.e. estimated position by previous HMM) and localization uncertainty U. We use uniform uncertainty on interval of size $F = 1 + 2 \cdot \lceil U/D^{'} \rceil$.

- A: $A = \{a_{ij}\}$ where $a_{ij} = P[q_{t+1} = S_j | q_t = S_i]$, $1 \leq i, j \leq N$: To take into account odometric uncertainty Δ for a displacement D, we defined A as:

$$a_{ij} = \frac{1}{\lceil 2\Delta/D' \rceil} rect_{\lceil \Delta/D' \rceil}(i - j - (\lceil D/D' \rceil)) \quad (2)$$

3.1.3 Observation matrix

The observation matrix $B = \{b_j(k)\}$, where $b_j(k) = P[O_k \ at \ t | q_t = S_j]$, $1 \leq j \leq N$ and $1 \leq k \leq M$ is the probability of observing O_k when location is S_j. The observation matrix B is computed from visual similarity between the M past observations $\mathbf{O}_k = \{O_{k-M+1}, ..., O_{k-1}, O_k\}$ and the set of potentially matching database images I_k associated to state/position S_j (Fig. 4). Visual similarity is a critical module for such solution. We propose to compute this probability from the similarity measure using the following formula:

$$b_j(k) = \alpha \cdot exp(-a \cdot \mathcal{D}_j^2(O_k, I_j)) \quad (3)$$

where a is a constant, $\mathcal{D}_j(O_k, I_j)$ is the local visual similarity measure for image I_j and α is a normalization con-

stant to impose $\sum_{j=1}^{j=N} b_j(k) = 1$. The determination of $\mathcal{D}_j(O_k, I_j)$ is explained in section 3.2.

A summary of the general estimation scheme is presented in algorithm 1.

Algorithm 1: Vision based absolute localization from odometric measurements and acquired images

Input: M last past observations $\mathbf{O} = O_{k-M+1}, ..., O_k$, M odometrics measurements $d_{k-M+1}, ..., d_k$, Estimated position of the vehicle $\tilde{S}_k$ and localization uncertainty U, Odometric uncertainty model to compute Δ, Geo-referenced database images I_j and associated metrics M_j.

Output: M corrected vehicle positions $\hat{\mathbf{S}} = \hat{S}_{k-M+1}, ..., \hat{S}_k$.

1 Compute $\mathbf{A}$ and $\mathbf{\Pi}$ from $\tilde{S}_k$, U and Δ and M as explained in section 3.1.2;
2 Select relevant geo-referenced database images from $\tilde{S}_k$, U, D, and M;
3 Compute similarities between the M past observations and relevant database images as explained in section 3.1.3;
4 Compute B from similarities with Eq.(3);
5 Apply Viterbi algorithm to solve Eq.(4) to estimate the latest vehicle position $\hat{S}_k$;

Given $\lambda = \{N, M, \Pi, A, B\}$, Eq. 4 can be solved.

$$\hat{\mathbf{S}}_k = \arg\max_{\mathbf{S}} P(\mathbf{O}_k|\mathbf{S}, \lambda) \cdot P(\mathbf{S}, \lambda)$$
$$= \arg\max_{\mathbf{S}} \left(\prod_{k=1}^{k=M} P(O_k|\mathbf{S}, \lambda) \right) \cdot \left(\pi_1 \cdot \prod_{k=2}^{k=M} a_{k-1,k} \right) \quad (4)$$

The first term of Eq. 4 refers to visual similarities between observations and the image database, whereas the second term refers to the dynamics of the vehicle and models spatio-temporal constraints.

3.2. Local visual similarity learning

We detail in this section our Exabal module that makes it possible to compute off line, i.e. in a pre-processing step, the local distance $\mathcal{D}_j$ that is required to compute Eq. 3.

We consider here the widely used Mahalanobis distance metric $\mathcal{D}_j = \mathcal{D}_{\mathbf{M}_j}$ that is parameterized by the positive semi-definite matrix (PSD matrix) $\mathbf{M}_j \in \mathbb{S}_+^d$ such that the distance between vectorial representations $(\mathbf{x}_j, \mathbf{x}_i) \in \mathbb{R}^d \times \mathbb{R}^d$ of the images (I_j,I_i) is written as follows:

$$\mathcal{D}_j^2(\mathbf{x}_j, \mathbf{x}_i) = \mathcal{D}_{M_j}^2(\mathbf{x}_j, \mathbf{x}_i) = (\mathbf{x}_j - \mathbf{x}_i)^\top \mathbf{M}_j (\mathbf{x}_j - \mathbf{x}_i) \quad (5)$$

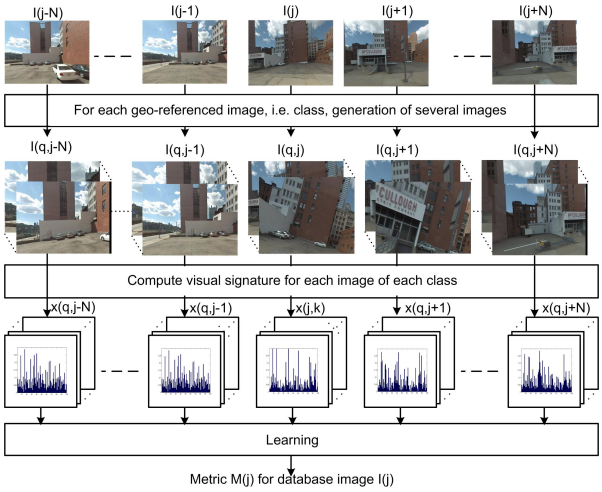


Figure 5. Creation of simulated similar and dissimilar examples in order to learn for each geo-referenced image database I_j a metric $\mathbf{M}_j$.

Exabal learns from artificial examples a local distance matrix $\mathbf{M}_j$, as defined in Eq. 5, for each image of the database I_j, leading to an exemplar-based metric learning scheme. The overall pipeline of the proposed metric learning scheme is described by algorithm 2 and the examples generation is illustrated by Fig. 5.

3.2.1 Exemplar-Based Constraints

Basically, we impose that the distance between each image signature $\mathbf{x}_j$ and other neighbor image signatures of the database $\mathbf{x}_{j'}$, $j' \neq j$, is larger than the distance between $\mathbf{x}_j$ and a query signature $\mathbf{x}_q$ representing the same scene as I_j. Images, which are representative of the unknown query images, are required during training. To this end, we propose to apply geometric and photometric transformations on I_j to generate proxies for potential test query images.

We note $T_s(\mathbf{x}_j)$ and $T_d(\mathbf{x}_{j'})$ the vectorial signature of, respectively, $T_s^{(i)}(I_j)$ and $T_d^{(i)}(I_{j'})$,where $T_s^{(i)}$ and $T_d^{(i)}$ are a set of image transformations. During training, we enforce the following constraints:

$$\mathcal{D}_{\mathbf{M}_j}(\mathbf{x}_j, T_d(\mathbf{x}_{j'})) \geq \mathcal{D}_{\mathbf{M}_j}(\mathbf{x}_j, T_s(\mathbf{x}_j)) + 1 \quad (6)$$

The constraints in Eq. 6 promotes matrices $\mathbf{M}_j$ that discriminate I_j images from $I_{j'}$ images, at the same time as taking into account potential transformations. An interesting property of our constraint generation approach is the ability to produce a large number of constraints by sampling different $T_s^{(i)}$ and $T_d^{(i)}$ transformations, making the optimization of $\mathbf{M}_j$ (with a potentially large number of parameters) robust to over fitting. In this paper, we focus on rotations and cropping operations, but some other modes of scene variations could be explored, as lighting, shadow and seasonal

variations.

Each image I_j is described by a feature $\mathbf{x}_j$ corresponding to a BOW vector [27] encoding spatial information [16]. We validate in the experiments (section 4) that the method learns a distance and selects discriminative and spatially localized features, making the similarity measure much more powerful than the distance in the input space.

3.2.2 Optimization

To minimize the number of misclassified constraints in Eq. 6, we introduce a standard hinge loss function ℓ_d for penalizing the violation of each constraint in Eq. 6: $\ell_d(\mathbf{x}_j, T_s(\mathbf{x}_j), T_d(\mathbf{x}_{j'})) = max\left[0, 1 - \left(\mathcal{D}_{\mathbf{M}_j}(\mathbf{x}_j, T_d(\mathbf{x}_{j'})) - \mathcal{D}_{\mathbf{M}_j}(\mathbf{x}_j, T_s(\mathbf{x}_j))\right)\right]$, as well as the following convex loss ℓ_s for each pair $(I_j, T_s^{(i)}(I_j))$: $\ell_s(\mathbf{x}_j, T_s(\mathbf{x}_j)) = \mathcal{D}_{\mathbf{M}_j}(\mathbf{x}_j, T_s(\mathbf{x}_j))$. ℓ_s aims at minimizing the distance between each image and its transformed version, *i.e.* between similar images. It can be interpreted as a regularization prior. It is to be noted that other regularization schemes could also be used, *e.g.* based on the Frobenius, nuclear norm [20] or methods giving an explicit control of the matrix rank as [15].

Our final objective $\mathcal{P}(\mathbf{M}_j)$ function combines the loss ℓ_s and ℓ_d over the whole set of constraints, with a weighting parameter μ:

$$\mathcal{P}(\mathbf{M}_j) = (1-\mu) \sum_{T_s \in \mathcal{T}} \ell_s(\mathbf{x}_j, T_s(\mathbf{x}_j)) + \mu \sum_{\substack{j' \neq j, \\ (T_d, T_s) \in \mathcal{T} \times \mathcal{T}}} \ell_d(\mathbf{x}_j, T_s(\mathbf{x}_j), T_d(\mathbf{x}_{j'})) \quad (7)$$

Eq. 7 is convex with respect to $\mathbf{M}_j$. We can use a stochastic projected gradient descent scheme to solve it. After each gradient computation, the matrix $\mathbf{M}_j$ is updated and projected onto the PSD cone if necessary. The algorithm is guaranteed to converge to the global minimum, up to a well-chosen gradient step. In practise, the optimization is fast with reasonable number of constraints and quickly converges.

3.2.3 Similarity measure within the HMM

Once a metric $\mathbf{M}_j$ is learnt for each image I_j of the database, it becomes possible for a given observation O_k to compute $b_j(k)$ thanks to Eq. 8, where $\mathbf{x}_k$ is the visual signature for observation O_k and $\mathbf{x}_j$ are the visual signatures of considered geo-referenced images.

$$b_j(k) = \alpha \cdot exp(-a \cdot (\mathbf{x}_j - \mathbf{x}_k)^T \mathbf{M}_j (\mathbf{x}_j - \mathbf{x}_k)) \quad (8)$$

When computing similarities with Eq. 8, there is no obvious guarantee that the different distances $D_{\mathbf{M}_j}(\mathbf{x}_j, \mathbf{x}_k)$ for different $\mathbf{M}_j$ are comparable, since each optimization has been performed independently. To alleviate this problem, we normalize each $\mathbf{M}_j$, as a post-processing learning step, so that the Frobenius norm of $\mathbf{M}_j$ is equal to 1. We could use more advanced normalization schemes as proposed in [10], but we found it sufficient in our experiments.

Algorithm 2: Local metric learning module

Input: Database geo-referenced image I_j
Output: Local metric M_j for geo-referenced database image I_j.

1. Select database geo-referenced neighbor images $I_{j\prime}$ ($j \neq j\prime$);
2. Generate simulated images by applying some transformations on images I_j and $I_{j\prime}$;
3. Compute visual signatures of all simulated images, i.e. $T_s(\mathbf{x}_j)$ and $T_s(\mathbf{x}_{j'})$;
4. From all visual signatures, learn metric M_j by optimizing cost function (Eq. 6).

4. Experimental results

4.1. Experimental setup

We built an image corpus from Google Pittsburgh dataset [5] for image database, and from Google Streetview images for query images [4]. These image dataset have been acquired at different time, resulting in strong visual changes for the same scenes (Fig. 2). Camera fields of view are also different. Pittsburgh dataset images have been resized to 640x480, so that their resolutions match the query image resolution. From the original corpus, we keep one image every $D' = 5m$ resulting in a corpus of 2215 images. Query images are downloaded from Google Streetview website (resolution of 640x480, field of view of 100^o, camera tilt of 5^o.) We requested one image every $D = 15m$ resulting in 846 query images.

BOW are computed from SIFT descriptors densely extracted [29]: four scales are used 1, 1.5, 2, 2.5 and the step between each SIFT descriptor is 4. BOW parameters are the followings: Hard assignment, Sum pooling, L2 Normalization, no tf-idf weighting. Spatial Pyramidal Matching configuration for BOW is 1x1, 2x2. The size of the codebook has been chosen to be 100.

Concerning the parameterization of our software, localization uncertainty U is set to $\pm 100m$, which is equivalent to an uncertainty error of ± 20 database image. Image retrieval is thus performed among N=41 database images. The odometric uncertainty is set to $\Delta = 10m$ for a mean displacement between two queries of about $15m$. The trade-off parameter μ between the two terms ℓ_s and ℓ_d of the objective function, was set to 0.5.

4.2. Results

We compared our solution (Exabal+HMM) including the similarity measure described in section 3.2 used within a HMM filter described in section 3.1 with two state of the art IR solutions. The first one is a similarity measure based on L2 norm between query and database BOW (L2+HMM), and the second one is based on a similarity computed thanks to the number of SIFT descriptors that matches between the query image and each database images after a RANSAC [9] geometric filter (Vote+HMM). We also report achieved performances when no HMM filter is used and for different BOW spatial configurations. Performances, i.e. mean localization error, IR accuracy an time to process one query, using the previously described setup, are given in Tab. 1.

These results show the interest of exemplar based local metric learning for visual geo-localization. Compared to (L2+HMM) method, our solution (Exabal+HMM) achieves a substantial accuracy gain of 8% (from 46% to 54%). At the same time, the mean localization error is reduced from 4.9m to 3.9m. Without HMM filter, achieved performances by our solution remains better than a standard IR solution based on BOW compared with Euclidean distance. In that case, Exabal improves performances by 8% (from 40% to 48%).

Achieved performances validate our objective function defined by Eq. 7, as well as the way we generate the constraints, *i.e.* by applying various transformations to database images in order to build representative images that likely look like to potential query images. The processing time is a slightly higher as the Mahalanobis distance is more complex to compute than an Euclidean distance, but at the same time it is less complex than a vote solution. Compared to a vote solution, the time to process one query (measured with Matlab) is roughly reduced by a factor 10.

Method	Spatial conf.	Loc. error	Acc.	Time
L2 + HMM	1x1	8.0m	42%	7.2s
	1x1, 2x2	5.3m	44%	
	1x1, 2x2, 1x3	4.9m	46%	
Exabal + HMM	1x1	4.5m	52%	8.6s
	1x1, 2x2	4.1m	54%	
	1x1, 2x2, 1x3	**3.9m**	**54%**	
Vote + HMM		4.1m	50%	55,6s
L2	1x1	17.4m	34%	6.8s
	1x1, 2x2	14.6m	38%	
	1x1, 2x2, 3x1	12.9m	40%	
Exabal	1x1	12.0m	42%	8.2s
	1x1, 2x2	11.3m	46%	
	1x1, 2x2, 1x3	10.1m	48%	
Vote		11.8m	50%	55,2s

Table 1. Mean localization error, accuracy and time to process one query with and without HMM for our solution compared to state of the art IR algorithms.

The use of the HMM reduces significantly the mean error localization, because HMM enables to filter out absurd match, i.e. those that do not respect spatio-temporal coherency of the query image sequence. It has to be noted that for the vote experiment, whereas mean error localization is reduced, IR accuracy is not improved: even if absurd matches are removed, correct matches are not found due to a bad similarity measure. The improvement provided by the HMM use confirms that exploiting the spatio-temporal constraint is essential. Furthermore, when the dimension of the visual signature increases (from 100 to 800 in our experiments), then results are slightly better, but this is not the main improvement factor.

Fig. 6 are some examples where (Exabal) without HMM filtering makes it possible to find the database image (b) that depicts the same scene as the query image (a), whereas (L2) solution can't (c).

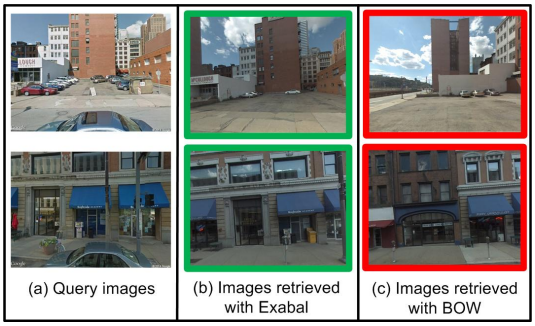


Figure 6. (a) Query images - (b) Images retrieved with (Exabal) solution - (c) Images retrieved with (IR-L2) solution.

4.3. Further insight

Thereafter, we analyze the reasons of performance improvements by (Exabal) only. We first illustrate that learnt metrics make it possible to retrieve the good image database if the query image is close to one of the generated image. Then, we show that improvements are due to the selection of discriminative features and the learning of invariance to photometric and geometric transformations.

- Learnt metrics during training

Fig. 7 shows the distances between all similar generated artificial BOW $T_s(\mathbf{x}_j)$ and $\mathbf{x}_j$ as well as $\mathbf{x}_{j'}$ BOW. The learnt metric enables to discriminate all potential generated query images from neighbor images, whereas it was impossible with the Euclidean norm. Indeed, whatever the query simulated image $T_s(\mathbf{x_j})$, $\mathcal{D}_{\mathbf{M}_j}(T_s(\mathbf{x_j}), \mathbf{x_j})$ remains inferior to $\mathcal{D}_{\mathbf{M}_j}(T_s(\mathbf{x_j}), \mathbf{x_{j'}})$. This means that if the signature of the effective query at test time is close to one of the generated

queries, then we will be able to retrieve the good image database thanks to the learnt metric.

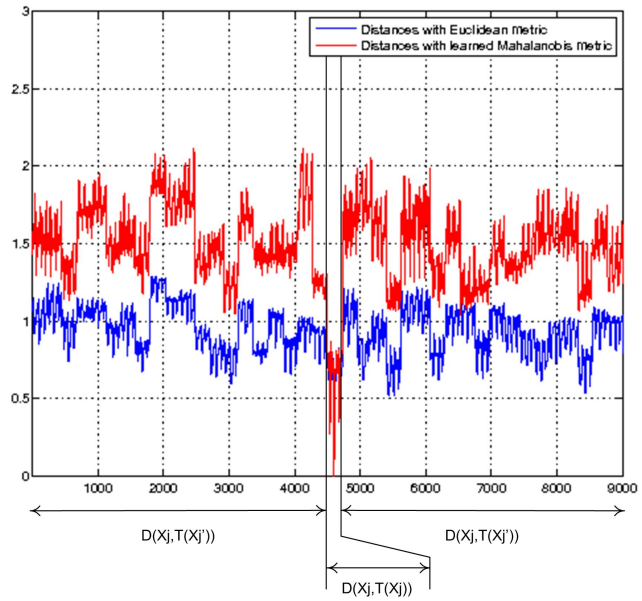


Figure 7. L2 and learnt Mahalanobis distances between potential query signatures $T_s(\mathbf{x}_j)$ and $\mathbf{x}_j$ signatures as well as $\mathbf{x}_{j'}$, i.e. signature of neighbor images $I_{j'}$ $(j' \neq j)$.

- Selection of discriminative features and discriminative image area(s)

The first advantage of our method is that it selects relevant, *i.e.* discriminative visual features. To visualize the most discriminative word for a given database image, we compute the eigenvector $\mathbf{v}_1$ of the largest eigenvalue λ_1 of $\mathbf{M}_j$ that represents the importance of each visual word. $\mathbf{M}^{'} = \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T$ is the nearest rank-1 matrix of $\mathbf{M}$ in the ℓ_2 norm. Thus $\mathcal{D}^2_M(x_j, x_k) \approx \lambda_1 (\mathbf{v}_1^T (x_j - x_k))^2$ and therefore $\mathbf{v}_1$ weights the importance of visual words. Fig. 8 shows the visual word having the highest value in vector $\mathbf{v}_1$. This visual word "window corner" makes it possible to retrieve the good image (b), although many features (the bricks of the wall) were common between the query image (a) and a neighbor image (c). What's more, as the BOW hold spatial information, we also check that Exabal is able to learn where are located the most discriminative features in the image.

- Learning invariance to photometric and geometric transformations

Another advantage of our method is that we learn invariance to photometric and geometric transformations. To demonstrate it, we selected randomly 1000 database images on which various transformations have been applied (*i.e.* various rotations from $-18°$ to $+18°$ around the 3 axes and various crops from 6 to 35 pixels). Thus we generate 10000 query test images $I_q = T_s^{(i)}(I_k)$, that have not been used during training. The mean classification rates for these simulated query images are reported in Tab. 2 for (L2) and (Exabal) solutions. A classification rate of more than 99% when learnt Mahalanobis distance is used confirms our claim concerning invariance to photometric and geometric transformations.

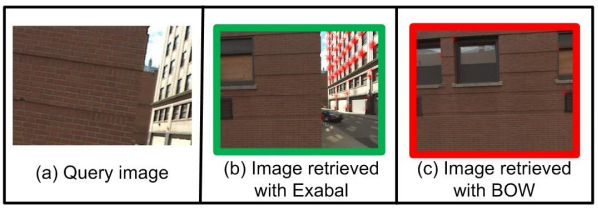


Figure 8. The visual word "window corner", that has been learnt to be discriminating, improves image retrieval task.

	Mean classification rates
L2	94.8%
Exabal	99.1%

Table 2. Mean classification rates computed for simulated test images.

5. Conclusion

We proposed a new visual geo-localization solution for autonomous vehicle adapted to applications for which geographical positions (GPS) of database images are available as well as an *a priori* approximate localization of the vehicle. The proposed solution includes exemplar-based learnt metrics within a HMM. Similarities are simple and fast to compute: for each observation, only one BOW has to be computed and N Mahalanobis distances. Learning method is generic: depending of the final application, transformations used during training can be easily adapted. We demonstrate that learnt similarities select discriminating features, and are able to gain invariance to meaningful transformations. We compared our framework with state of the art image retrieval algorithms evaluated on a corpus of 846 queries and 2215 database images. Our solution improves accuracy from 40% for a traditional BOW solution to 54%, while maintaining the same processing time. At the same time, the mean localization error is reduced from 12.9m to 3.9m. Finally, even if we use BOW as visual signature, other visual signatures can be easily used, as the recently deep features [25] which have been demonstrated to be efficient.

6. Acknowledgment

This work results from a collaboration between UPMC University, Onera and Thales Services SAS.

References

[1] S. Avila, N. Thome, M. Cord, E. E. Valle, and A. D. A. AraúJo. Pooling in image representation: The visual codeword point of view. *Computer Vision and Image Understanding*, 117(5):453–465, 2013.

[2] H. Badino, D. Huber, and T. Kanade. Real-time topometric localization. In *Proceedings of the International Conference on Robotics and Automation*, pages 1635–1642. IEEE, 2012.

[3] M. Brubaker, A. Geiger, and R. Urtasun. Lost! leveraging the crowd for probabilistic visual self-localization. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 3057–3064, June 2013.

[4] G. company. Google street view API. `http://developers.google.com/maps/documentation/streetview`.

[5] G. company. Pittsburgh dataset provided by google for research purposes. `http://www.icmla-conference.org/icmla11/challenge.html`.

[6] M. Cummins and P. Newman. Fab-map: Probabilistic localization and mapping in the space of appearance. *International Journal of Robotics Research*, 27:647–665, June 2008.

[7] M. Cummins and P. Newman. Appearance-only slam at large scale with fab-map 2.0. *International Journal of Robotics Research*, 30:1100–1123, Aug. 2011.

[8] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. A. Efros. What makes paris look like paris? *ACM Transactions on Graphics (SIGGRAPH)*, 31(4), 2012.

[9] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24:381–395, June 1981.

[10] M. Gebel and C. Weihs. *Calibrating classifier scores into probabilities*. Advances in Data Analysis. Springer Science and Business Media, 2007.

[11] P. Gronat, G. Obozinski, J. Sivic, and T. Pajdla. Learning and calibrating per-location classifiers for visual place recognition. In *Proceedings of the Computer Vision and Pattern Recognition conference*, pages 907–914, June 2013.

[12] J. Hays and A. Efros. Im2gps: estimating geographic information from a single image. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*. IEEE, 2009.

[13] H. Jegou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid. Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1704–1716, Sept. 2012.

[14] J. Knopp and J. S. T. Pajdla. Avoiding confusing features in place recognition. In *Proceedings of the European Conference on Computer Vision*, volume 6311, pages 748–761. Springer, 2010.

[15] M. Law, N. Thome, and M. Cord. Fantope regularization in metric learning. In *Proceesdings of the international conference on Computer Vision and Pattern Recognition*, 2014.

[16] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bag of features: spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the Computer Conference on Computer Vision and Pattern Recognition*, pages 2169–2178, June 2006.

[17] C. LeBarz, N. Thome, M. Cord, S. Herbin, and M. Sanfourche. Global robot ego-localization combining image retrieval and hmm-based filtering. In *6th workshop on Planning Perception and Navigation for Autonomous Navigation*, Sept. 2014.

[18] W. Maddern, M. Milford, and G. Wyeth. Cat-slam: Probabilistic localisation and mapping using a continuous appearance-based trajectory. *International Journal of Robotics Research*, 31(4):429–451, Apr. 2012.

[19] A. Majdik, Y. Albers-Schoenberg, and D. Scaramuzza. Mav urban localization from google street view data. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 3979–3986, Nov. 2013.

[20] B. McFee and G. Lanckriet. Metric learning to rank. In *ICML*, 2010.

[21] C. McManus, B. Upcroft, and P. Newmann. Scene signatures: Localised and point-less features for localisation. In *Proceedings of Robotics: Science and Systems*, Berkeley, USA, July 2014.

[22] M. Milford and G. Wyeth. Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In *Proceedings of the International Conference on Robotics and Automation*, pages 1643–1649. IEEE, May 2012.

[23] E. Pepperell, P. Corke, and M. Milford. All-environment visual place recognition with SMART. In *Proceedings of the International Conference on Robotics and Automation*, pages 1612–1618, 2014.

[24] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77, pages 257–286, 1989.

[25] S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf : An astounding baseline for recognition. In *Proceedings of Computer Vision and Pattern Recognition conference*, 2014.

[26] G. Schindler, M. Brown, and R. Szeliski. City scale location recognition. In *Proceedings of the Computer Vision and Pattern Recognition conference*, pages 1–7, June 2007.

[27] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 1470–1477, Oct. 2003.

[28] G. Vaca-Castano, A. Zamir, and M. Shah. City scale geospatial trajectory estimation of a moving camera. In *Proceedings of the Computer Vision and Pattern Recognition conference*, pages 1186–1193, 2012.

[29] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008.

[30] A. Zamir and M. Shah. Accurate image localization based on google maps street view. In *Proceedings of the European Conference on Computer Vision*, pages 255–268, Sept. 2010.

[31] J. Zhang, A. Hallquist, E. Liang, and A. Zakhor. Location-based image retrieval for urban environments. In *Proceedings of the International Conference on Image Processing*, pages 3677–3680. IEEE, 2011.