Pairwise Probabilistic Voting: Fast Place Recognition without RANSAC

Edward David Johns and Guang-Zhong Yang

Imperial College London, UK

Abstract. Place recognition currently suffers from a lack of scalability due to the need for strong geometric constraints, which as of yet are typically limited to RANSAC implementations. In this paper, we present a method to successfully achieve state-of-the-art performance, in both recognition accuracy and speed, without the need for RANSAC. We propose to discretise each feature pair in an image, in both appearance and 2D geometry, to create a triplet of words: one each for the appearance of the two features, and one for the pairwise geometry. This triplet is then passed through an inverted index to find examples of such pairwise configurations in the database. Finally, a global geometry constraint is enforced by considering the maximum-clique in an adjacency graph of pairwise correspondences. The discrete nature of the problem allows for tractable probabilistic scores to be assigned to each correspondence, and the least informative feature pairs can be eliminated from the database for memory and time efficiency. We demonstrate the performance of our method on several large-scale datasets, and show improvements over several baselines.

Keywords: Place Recognition, Location Recognition, Instance Recognition, Image Retrieval, Bag Of Words, Inverted Index.

1 Introduction

This paper addresses place recognition [6][16][8][26], whereby the identity is sought of a particular place or scene depicted in a query image. This is closely related to the recognition or retrieval of object instances [13][1] and qualitative localisation [7][17][2]. In a typical place recognition framework, a Bag Of Words (BOW) filtering stage yields a subset of candidate images [14], upon which strong 3D constraints are imposed on local features based on a RANdom SAmple Consensus (RANSAC) scheme [15]. Whilst such constraints offer powerful verification of global geometric consistency, they are expensive to compute due to the need for generating multiple transformation hypotheses in the RANSAC algorithm, and large-scale real-time recognition tasks are often forced to forego 3D geometry entirely [2][7]. Although the use of an inverted index in the BOW stage allows for very fast pre-filtering and has been studied extensively in recognition and retrieval methods [5][1], geometric verification is typically reserved as a separate stage altogether [14], or weakly represented in the BOW vector [4], but typically is not incorporated directly in the inverted index.

D. Fleet et al. (Eds.): ECCV 2014, Part II, LNCS 8690, pp. 504–519, 2014.

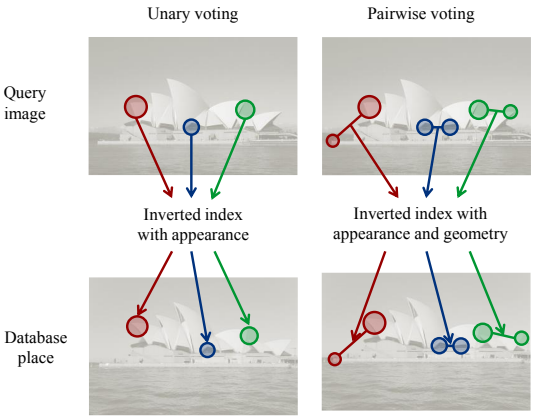


Fig. 1. A comparison of our proposed pairwise voting method with the standard single feature voting method, both using an inverted index

In this paper, we consider an alternative approach to geometric matching by building a generative model for each place, and embedding the learned 2D pairwise geometry directly in the inverted index to allow for much faster querying, which we denote *Pairwise Probabilistic Voting* (PPV). First, local features are tracked across training images for each place using wide baseline matching. Then, the relative geometry of pairs of features is discretised into a dictionary of *geometric words*, and the informative elements of an image are taken as the two visual words and one geometric word that defines each feature pair, which we denote a *word triplet*. Given a query image, rather than searching for candidate places that contain a particular visual word as with standard BOW, the search is for places that contain a particular pairwise configuration satisfying all three words in the triplet, as illustrated in Figure 1. By learning a probabilistic model of local geometry for each pair of features across training images for a particular place, votes for each place are weighted with a score reflecting the likelihood that the pairwise correspondence is a true positive. Finally, by ensuring that all votes for a candidate place agree globally with each other in pairwise geometry, an approximate constraint on global geometry can be applied without the need for an expensive RANSAC step.

1.1 Related Work

Instance recognition typically extends the image retrieval framework [14] to a recognition framework [26], by exploiting structure in the database [3] and learning models of places or objects over a training set of training images. Standard image retrieval approaches based on ranking database images form a simple solution [13], and in [16] a query was matched to each of a small subset of exemplar images for each place. Tracking local features across several training images to learn the expected behaviour of features was proposed in [6], and superimposing

them onto a single synthetic image was investigated in [8]. Competing ideologies to recognition include learning discriminative models based on the BOW vector [9] and building point clouds from which to draw feature correspondences [17].

Although geometric matching for recognition with these techniques still relies on a costly RANSAC iteration, attempts to enrich this stage using 2D geometries have proven successful in image retrieval. In [22], matches were made between groups of neighbouring features that were weakly spatially consistent, and in [25][24] dictionaries combining both appearance and geometry were learned. In [20], fast Hough-based voting based on feature geometries was proposed, and [19][18] developed techniques for matching only those subsets of features with agreement on global transformations across an image pair. Near-duplicate image search methods also allow for enforcement of strong local geometric constraints [23], but these are not suitable for wide baseline matching or outdoor scenes.

2 A Geometric Dictionary

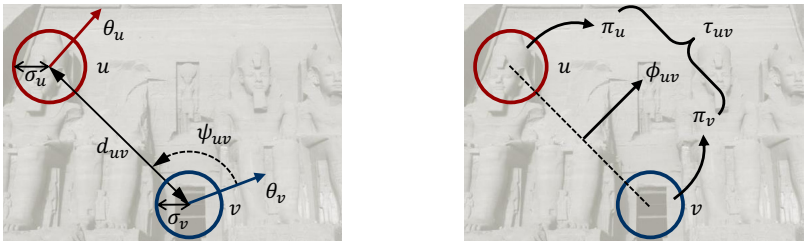
Concurrently with a standard visual dictionary [14], we propose to discretise pairwise geometry to enable geometric data to be sent through an inverted index alongside the appearance data. As with the visual dictionary Π of visual words $\pi \in \Pi$, we define a *geometric dictionary* Φ of *geometric words* $\phi \in \Phi$. Each geometric word represents a unique range in pairwise 2D image space.

The geometric dictionary is composed over 4-dimensional space, with each geometric word defined by 4 pairwise geometries. For a feature pair constituting features u and v, the pairwise geometries are as follows: scale-invariant distance δ_{uv}, scale ratio σ_{uv}, rotation-invariant orientation difference θ_{uv} and rotation-invariant angle ψ_{uv}. Scale-invariance and rotation-invariance are important in ensuring that the generative model of each place is not limited by the scales and rotations reflected in the training dataset. The distance between features, d_{uv}, is made scale-invariant by dividing by the scale of u, and the relative angle is made rotation-invariant by subtracting the orientation of u. The scale ratio and orientation difference are naturally invariant to in-plane transformation. This discretisation of geometry is similar to that of [25] except for we define the discretisation a priori rather than during querying, and is comparable to [24] but we limit to pairwise geometries to allow for inverted indexing with reasonable memory requirements.

Using the notation in Figure 2, the pairwise geometries are then calculated as follows:

$$\begin{aligned} \delta_{uv} &= \frac{d_{uv}}{\sigma_u}, & \sigma_{uv} &= \frac{\sigma_v}{\sigma_u}, \\ \theta_{uv} &= \theta_v - \theta_u, & \psi_{uv} &= \psi_{uv} - \theta_u \end{aligned} \tag{1}$$

Each of these four pairwise geometries is independently discretised by defining boundaries for each of the four geometries, with n divisions per geometry. However, rather than defining each boundary as a linear function of n, we instead use the expected distribution of each geometry from a training set of feature pairs,



(a) Each geometric word ϕ is defined by the above geometries

(b) Each word triplet τ is defined by two visual words π_u, π_v and one geometric word ϕ_{uv}

Fig. 2. Notation for pairwise geometries

and compute the k^{th} percentile from that distribution. This is because whilst the orientation and angle differences can be assumed to be distributed uniformly in the range $0 - 360°$, there is no such trend in the distribution of pairwise scale ratios and distances. By considering the observed distribution over a set of feature pairs from real data, this ensures that for a geometric dictionary with a fixed number of divisions per geometry, each division has an equal likelihood of assignment given a new pair of features, tending towards a uniform global distribution. We learned these distributions by randomly sampling one million pairs of features observed in images from the database.

Finally, the geometric word ϕ_{uv} for the feature pair is the discrete portion of 4-dimensional geometry defined by the quantised values of δ_{uv}, σ_{uv}, θ_{uv} and ψ_{uv}. This geometric word is then combined with the visual words π_u and π_v of the two features, to form a word triplet τ_{uv}. For our experiments, we used 30 divisions for each of the four geometries, yielding a geometric dictionary of size $810K$.

3 Pairwise Probabilistic Voting

The key idea behind our proposed method is to create a generative model for each place in the database, by learning distributions of word triplets over the place's training images, and then finding matches between pairs of query features and pairs of database landmarks. These matches then vote for the respective place, with the vote weighted probabilistically.

3.1 Learning a Distribution of Triplets

Let us define a *landmark* as a real-world point in the environment, that is observed in an image as a feature. Every database place $s \in \mathcal{S}$ is represented by a set of such landmarks $\mathcal{X}_s$, with each built from a single feature track across

the place's entire set of training images using wide baseline matching, similar to [6][8][16]. Each pair of features u and v, forming two tracks, is assigned one word triplet per image in which the features co-occur, and thus a set of word triplets is accumulated over all training images. The two tracks form two landmarks x and y, with the landmark pair z_{xy} assigned the distribution of triplets $p(\boldsymbol{\tau}|z_{xy})$ based on this learned set of word triplets from the feature tracks.

3.2 Voting

For the remainder of the paper, we drop the subscripts in z and let it represent any particular landmark pair. During recognition, each feature pair w in the query image is assigned a single word triplet τ_w. These query word triplets are then sent through an inverted index to find landmark pairs in the database that have also been assigned to this particular word triplet. The leaf node in the index tree assigns a weighted vote μ_z to landmark pair z, and the overall score for a place s is then the normalised summation of votes across all its landmark pairs:

$$f(s) = \frac{\sum_{z \in \mathcal{Z}_s} \hat{\mu}_z}{\eta} \tag{2}$$

Here, $\mathcal{Z}_s$ is the set of landmarks in place s, and η is a normalisation term, defined as the average number of landmark pairs observed in s's training images. $\hat{\mu}_z$ is defined as the maximum value of μ_z achieved by all query feature pairs, to account for cases when more than one feature pair matches a landmark pair.

The weighted vote μ_z is a probabilistic score representing how likely it is that observed triplet τ_w is a true observation of landmark pair z. From the learned distribution $p(\boldsymbol{\tau}|z)$, we can readily draw the value of $p(\tau_w|z)$. Furthermore, we can also draw the values of $p(\tau_w|z^*)$ for any landmark pair z^* in the entire database. Therefore, for a given triplet τ_w, the vote for landmark pair z is evaluated as:

$$\mu_z = p(z|\tau_w) = \frac{p(\tau_w|z)p(z)}{\sum_{z^*} p(\tau_w|z^*)p(z^*)} \tag{3}$$

The value of $p(\tau_w|z)p(z)$ is proportional to the number of times τ_w is observed for landmark pair z across all training images. Therefore, μ_z is simply the number of times that τ_w is observed when z is present, divided by the number of times that τ_w is observed when any landmark pair is present. This weight is calculated in advance and stored at the leaf node in the index, such that voting for a place involves simply traversing the index with a query triplet and adding weighted votes to any database places, should a landmark pair for that place have an entry at that leaf node. Figure 3 illustrates a set of landmark pairs and their associated weights. Higher weights are assigned to pairs which are both frequently observed at a place and discriminative with respect to all other landmark pairs in the database.

Fig. 3. Vote weights μ_z for landmark pairs for an example place. The blue circle represents landmark x, with all other circles representing landmark y, and hence a landmark pair z_{xy}.

3.3 Index Structure

Our goal in designing an appropriate index is to link every word triplet to a leaf node, with each leaf node pointing to a set of weighted votes, and each one associated with a database place. The fastest inverted index is one which contains one pointer for every possible word triplet, where each pointer represents the leaf node. However, given a visual dictionary containing $100K$ words and a geometric dictionary containing $800K$ words,, this would require an index with at least $100K \times 100K \times 800\text{K} \approx 10^{16}$ pointers, which is impractical.

We therefore propose to divide the inverted index into two layers, each with a different structure. The first layer is a standard inverted index, and represents the combination of the two visual words in a word triplet, requiring $100K \times 100K = 10^{10}$ pointers (10 GB RAM). Then, the second layer is no longer an inverted index structure, but simply a list of geometric words for each visual word combination from the first layer. This list represents all geometric words from the entire database that have been assigned to by this visual word combination, and the list is structured as a binary search tree for efficient searching. In this way, whilst the first layer contains an entry for every possible combination of visual word pairs, the second layer only contains those geometric words which have actually been observed together with the particular visual word pair, reducing the memory requirements by several orders of magnitude.

Figure 4 illustrates this structure. In this example, suppose we have a query triplet consisting of visual words π_2 and π_3, and geometric word ϕ_4. Following the path of π_2 for the first landmark and then π_3 for the second, we see the list of two geometric words: ϕ_1 and ϕ_4. Then, taking ϕ_4, we see that there are two votes, targetting two different landmark pairs in the database that are represented by this particular word triplet, and we then add these weighted votes to scores for the respective places.

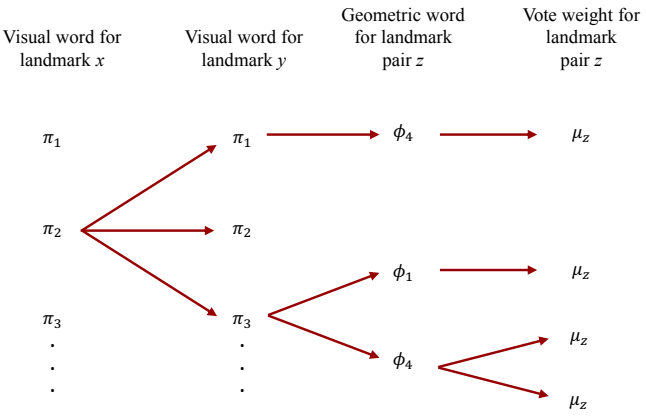


Fig. 4. The index structure, with an example distribution of wods. The arrows represent potential paths through the index in pursuit of the weighted votes.

4 Smoothing the Distributions

The discrete nature of the generative place models makes it tractable to learn a deep probabilistic model of landmark pairs, as a joint distribution across both appearance and geometry. Rather than treat the visual and geometric words assigned to a pair as independent, we propose to compute a full joint distribution across all three words in a triplet. In this way, effects on one word are modelled in the knock-on effect on another word. For example, if a particular illumination condition causes the visual word of one landmark to change, then we model the corresponding effect of this illumination condition on the visual word of the other landmark in the pair. Similarly, if the viewpoint on the place changes such that the pair's geometric word is affected, then the effect on the visual words due to the apparent change in scene appearance can be modelled. This involves updating the distribution $p(\boldsymbol{\tau}|z)$ for each landmark pair by smoothing the explicit observations of word triplets from the feature tracks in the training images.

For an observed visual word $\pi \in \Pi$, let us define a set of a_π *alternative visual words*, $\bar{\pi} \in \Pi$, where an alternative word represents a possible assignment on a subsequent observation of the same landmark. This is similar to the soft assignment strategies in the BOW framework [13][11]. Each alternative visual word is designated a likelihood $p(\bar{\pi}_x|\pi_x)$ following Gaussian weighting proportional to the word centroid distance, with standard deviation σ_π, as in [13]. Similarly, for each geometric word $\phi \in \Phi$, a set of a_ϕ *alternative geometric words* $\bar{\phi} \in \Phi$ are defined as the α_ϕ nearest geometric words to ϕ, with their likelihoods $p(\bar{\phi}|\phi)$ again weighted by a Gaussian, with standard deviation σ_ϕ. We chose the values

for a_π and a_ϕ by observing sets of landmark pairs, and determining the limit when the set of alternative words assigned to the pair accounted for future observations in 99% of the cases. The values of σ_π and σ_ϕ were fixed such that the furthest away alternative word had a value of $p(\bar{\pi}|\pi)$ or $p(\bar{\phi}|\phi)$ at 1% of that of the original word, i.e. when $\bar{\pi} \equiv \pi$) or $\bar{\phi} \equiv \phi$).

We now consider the probability that an observation of landmark pair z is assigned to a particular word triplet τ. Let us factorise the distribution as follows:

$$\begin{aligned} p(\tau|z) &= p(\pi_x, \pi_y, \phi_{xy}|z) \\ &= p(\pi_x|z)p(\pi_y|\pi_x, z)p(\phi_{xy}|\pi_x, \pi_y, z) \end{aligned} \tag{4}$$

As such, we model y's visual word to be dependent on x's visual word, and the geometric word as dependent on both these visual words. We now introduce the smoothing effects of the alternative words, by considering that new word triplets should be included in the distribution, if each word in the triplet is in fact an alternative word for the respective original word. The probability of observing this new triplet is then calculated by taking the factorisation in Equation 4, and replacing each term with the probability of assignment to the alternative word. This is evaluated as:

$$\begin{aligned} p(\tau|z) &= p(\pi_x, \pi_y, \phi_{xy}|z) \\ &\quad \text{contribution from alternative visual words for landmark } x \\ &= \overbrace{\sum_{\bar{\pi}_x} p(\bar{\pi}_x|\tau)p(\bar{\pi}_x|\pi_x)} \\ &\quad \text{contribution from alternative visual words for landmark } y \\ &\times \overbrace{\sum_{\bar{\pi}_y} p(\bar{\pi}_y|\bar{\pi}_x)p(\bar{\pi}_y|\pi_y)} \\ &\quad \text{contribution from alternative geometric words for landmark pair } z \\ &\times \overbrace{\sum_{\bar{\phi}_{xy}} p(\bar{\phi}_{xy}|\bar{\pi}_x, \bar{\pi}_y)p(\bar{\phi}_{xy}|\phi_{xy})} \end{aligned} \tag{5}$$

where the probabilities in the bottom three rows are based on the maximum-likelihood distributions as before.

5 Geometric Cliques for Global Consistency

Whilst the pairwise geometry embedded in the inverted index offers strong constraints on local configurations, as of yet there is no enforcement of global geometric consistency. Thus, a set of feature pairs voting for one place may be independently representative of a landmark pair, but when considering the global relationships between all pairs, the overall configuration may be incompatible.

5.1 Defining a Compatibility Matrix

The proposed solution, which we denote the method of *Geometric Cliques* (GC), is based on finding a maximum clique in an adjacency matrix, whose elements

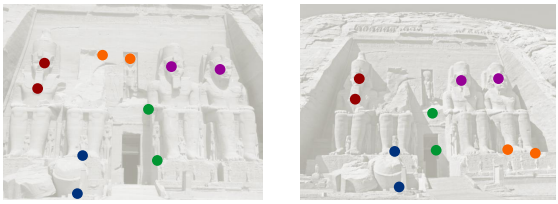
indicate the *compatibility* of each pairwise match. Let us define a set of n pairwise matches $m \in M$ between an image and a place, generated by passing query feature pairs through the inverted index. Then, for each place, we construct an $n \times n$ binary *compatibility matrix* B, where element B_{ij} stores the compatibility of pairwise matches m_i and m_j, and is set to either 0 or 1. If m_i represents a match between feature pair $w_{u^i v^i}$ and landmark pair $z_{x^i y^i}$, the value of B_{ij} is 1 if, and only if, there also exist pairwise matches between every feature pair $w_{u^i u^j}$, $w_{u^i v^j}$, $w_{v^i u^j}$, $w_{v^i v^j}$ and every landmark pair $z_{x^i x^j}$, $z_{x^i y^j}$, $z_{y^i x^j}$, $z_{y^i y^j}$. In other words, all pairwise combinations of features and landmarks in m_i and m_j must have found a match through the inverted index for m_{ij} to be set to 1. If any pair of these features has not found a match to any pair of these landmarks, then m_i and m_j are not fully compatible and B_{ij} is set to 0. In the case that a pair of landmarks never co-occur in the place's training images, then the respective element of B is always set to 1, i.e. a value of 0 indicates that we have explicitly observed an inconsistent pairwise match that needs to be eliminated.

For example, in Figure 5a, the left image can be considered the query features, and the right image the database landmarks. Taking the red and blue pairs, all four features in the left image have a geometrically-similar configuration to the associated landmarks in the right image. Therefore, the respective element in B is set to 1 (see the first matrix in Fig 5b). However, taking the red and orange pairs, the configuration is not similar across all four features (only across the two features highlighted by the same colour), and hence the respective element in B is set to 0.

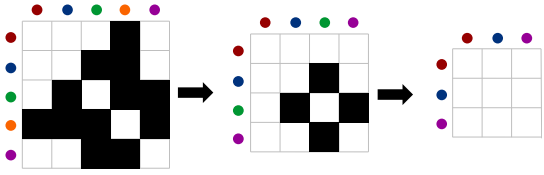
5.2 Searching for the Maximum Clique

The task now becomes to find a set of pairwise matches that are all compatible with each other, i.e. finding the maximum clique of B. Several solutions to this exist, including fast branch-and-bound methods [12], or approximate solutions using a fast search for a near-optimal maximum clique, followed by gradient descent to avoid local minima [21]. However, these methods are generalised and deal with a wide range of matrix structures, whereas we now propose our own fast approximate solution that exploits the unique nature of B.

There is a very low probability that both query features in a false positive pairwise match are also compatible with other pairwise matches. Thus, any false positive matches in B will have a very sparse row in the matrix, typically with very few elements set to 1. However, true positive pairwise matches will have a much larger number of compatible pairs, and hence the corresponding row will have a significant number of 1's. As such, we can very quickly eliminate false positives by detecting those rows with few 1's. The proposed algorithm exploits this by scoring each pairwise match by the number of consistent pairs, i.e. the number of elements in the respective row of B assigned to 1, and recursively removing the pair with the lowest score. After each iteration, the scores for each remaining pair are updated. The algorithm then converges when the entire matrix is devoid of any 0's.



(a) Each colour represents one pairwise match found through the inverted index, and the resulting adjacency matrix is shown in (b)



(b) The adjacency matrix iteratively eliminates the pairwise match with the lowest score, until a maximum clique is found

Fig. 5. Illustration of the Geometric Cliques algorithm to eliminate pairwise matches that, although may be locally consistent, are not globally consistent with other pairwise matches

See Figure 5b for an example of the evolution of B towards a maximum clique. Here, the green pairwise match is in fact a false positive, even though it may appear consistent with some of the other pairwise matches; in this example, the red pair. Our method is able to deal with this case because we require every pair to be consistent with every other pair in the final matrix.

Once the maximum clique has been established, the score for the respective place is determined in Equation 2 by considering only those pairs in the maximum clique. Given that false positive candidates exhibit zero or very few 1's in their respective row and hence are easy to eliminate, this method reproduced the same maximum clique as the methods of [12] and [21] over all experiments presented in Section 7, but with an average speed up factor of 13.7 compared with the fastest of these two.

During recognition of a query image, database places are ranked in order of their current scores, before elimination of these false positive matches as discussed. First, the place with the highest score is processed with geometric cliques, with its score updated, then the place with the second highest score, and so on. The algorithm stops when the score of the next place is lower than the maximum updated score for places which have been through the geometric cliques stage, because it is only possible for the score to reduce. Hence, only a small fraction of database places need to go through this stage, offering significant speed-up compared to RANSAC-based re-ranking strategies.

6 Boosting Efficiency by Triplet Selection

A key requirement of modern retrieval and recognition engines is scalability to web-scale tasks, whereby databases are automatically generated from vast user-generated databases [26][3]. We now explore how our proposed system scales appropriately to such demand, in terms of both memory efficiency and computational efficiency.

In order to both reduce the memory footprint and increase the recognition speed, albeit at a small cost to recognition accuracy, only a subset of all word triplets are stored in the index, with the rest discarded. Memory requirements will naturally be reduced, and meanwhile, speed will be increased when attempting to find a particular geometric word in the binary tree search, as each binary tree will be reduced in length. We propose to store only the most informative word triplets, such as those representing landmark pairs that are very stable in a place, or landmark pairs which have word triplets particularly unique to their place. Note that the emphasis is on informative word triplets, not informative landmark pairs; the triplets associated with each landmark pair will differ in their own discriminative power due to the number of other landmark pairs represented by each triplet.

To determine the level of information which each word triplet conveys, we consider the conditional entropy of the place identity, given that the knowledge of the triplet's presence or absence in a query image is available. Let us define $\mathbf{S}$ as the state of the place depicted in the query, which can take on all values $s \in S$, and the binary variable $\mathbf{T}$ as the state of word triplet τ, where $\mathbf{T} = 1$ indicates that the triplet is observed in a query. Word triplets are then ranked in order of the conditional entropy:

$$\begin{aligned} H(\mathbf{S}|\mathbf{T}) &= \sum_{\mathbf{S}\in S} \sum_{\mathbf{T}=0,1} p(\mathbf{S},\mathbf{T}) \log \frac{p(\mathbf{T})}{p(\mathbf{S},\mathbf{T})} \\ &= \sum_{\mathbf{S}\in S} \sum_{\mathbf{T}=0,1} p(\mathbf{T}|\mathbf{S}) \log \frac{p(\mathbf{T})}{p(\mathbf{T}|\mathbf{S})} \end{aligned} \tag{6}$$

where the second row comes from $p(\mathbf{S},\mathbf{T}) = p(\mathbf{T}|\mathbf{S})p(\mathbf{S})$, where the terms $p(\mathbf{S})$ then cancel due to an equal prior probability across all places.

To calculate $p(\mathbf{T} = 1|\mathbf{S} = s)$, the probability of observing word triplet τ given place s, we consider the proportion of s's training images that contain a landmark pair with this triplet, based on the landmark's distribution of word triplets. The value of $p(\mathbf{T} = 0|\mathbf{S} = s)$ is then $1 - p(\mathbf{T} = 1|\mathbf{S} = s)$, and the value of $p(\mathbf{T})$ is the summation of $p(\mathbf{T}|\mathbf{S})$ over all places. In order to choose an optimal set of word triplets for a specified memory constraint, triplets are added to the inverted index in order of their conditional entropy, such that those which offer most information when observed, are added first. This is evaluated in Section 7 for a given memory allowance. Note that the empty index, before any triplets are added, is a constant for any scale of database.

7 Experiments

7.1 Experimental Procedure

We evaluated our method on three datasets: the *Oxford* [14] and *Paris* [13] Buildings, and our own new dataset *World* Buildings. The Oxford and Paris Buildings datasets consist of 17 and 12 places respectively, and the World Buildings dataset consists of 300 places acquire from Flickr.com using search terms such as "Sydney Opera House" and "Houses of Parliament", each with 1000 training images and 10 test images. It was decided to use this new dataset due to the large number of individual places compared to standard image retrieval datasets such as the Oxford and Paris Buildings, together with the large number of training images as is required for our method. SIFT features [10] were matched using fast geometric matching [14] to generate feature tracks, whilst discarding tracks between image pairs yielding less than 15 inlier feature matches. For each dataset, a further 1M random distractor images were added from Flickr, with each image acting as its own place, such that each feature is designated a landmark, and with the probabilistic model for PPV computed across the single image. A dictionary of 100k visual words was trained using approximate k-means [14].

Our PPV method was compared against implementations of two modelling techniques, each with three geometric querying methods, for a total of 6 competitors. The modelling techniques include the Iconic Images (IC) method [16], returning the image with the most inliers across a set of iconic images for each place, and the Scene Maps (SM) method [8], with each place represented by a single map of superimposed features and returning the scene map with the most inliers. The querying techniques include the Visual Phrases (VP) of [25] where small groups of geometrically-consistent features are voted for, and the spatially-constrained similarity measure (SCSM) of [18] where entire object transformations are voted for, both using an inverted index. Furthermore, a standard technique was implemented using RANSAC geometric verification on the top 50 places returned from a BOW stage (BOW + RANSAC) [14], using the recently-updated LO-RANSAC method [15]. For each implementation, both the Average Precision (AP) and the Recall at 100% precision (R@1) were recorded, with the latter being a useful measure for the applicability to robotics due to the need for very high precision in localisation.

For a fair comparison of scalability, each modelling technique was allocated the same memory to store the necessary data (excluding constant memory requirements such as the BOW and word triplet index structures, which are not affected by scale). For II, iconic images were added in order of their distance to the centroid of the place's training images until the memory limit was reached, with all these iconic images then stored in the database. Similarly, for SM, the superimposed features were added in order of the number of features in the respective feature track. For our PPV method, word triplets were added in order of their conditional entropy as in Section 6. Feature attributes (location, scale and orientation) were quantised to 2 bytes each for SM and II.

7.2 Results

Accuracy and Timing. Table 1 presents the average precision, recognition rate and query time (excluding feature extraction and quantisation) for all implementations of competing methods, each given an allocation of 16MB (excluding the distractor images). Furthermore, we show the incremental improvements on our PPV method as two key components are introduced: the smoothed parameter learning stage (as opposed to maximum-likelihood estimation), and the geometric cliques stage (as opposed to a purely local voting scheme). This new method outperforms all other competitors that forego a RANSAC stage, and offers similar if not greater recognition performance even when RANSAC is included in the competitors (but not in our method). Intuitively, our geometric cliques stage for global geometric verification is very powerful and offers similar constraints to a full RANSAC procedure. Furthermore, our method is significantly faster than all competitors on all datasets, and in particular, adding the GC stage incurs little timing penalty compared to the addition of a RANSAC stage for the competitors. Note that performance is generally higher on the Oxford and Paris datasets despite the smaller set of training images for each place due to the availability of bounding boxes, whereas the World Buildings dataset requires unsupervised feature tracking across the entire image.

Memory. Figure 6 demonstrates that as further triplets are accumulated in the index, the recognition accuracy improves because a larger number of votes are possible for the correct place, and so this score becomes less corrupted by competing places. In fact, average precision begins to flatten out at around 8MB, corresponding to less than half of the maximum memory requirement (16MB), showing that half of the word triplets offer little information and are typically drawn from unstable landmark pairs, or those landmark pairs with poorly discriminating visual word and geometric word combinations. As a further

Table 1. Summary of recognition results for all implementations. AP = Average Precision, R@1 = Recall at 100% precision. (Time in ms, 3.2GHz Intel Core i7).

Method	Oxford			Paris			World		
	AP	R@1	Time	AP	R@1	Time	AP	R@1	Time
II + VP	0.673	0.695	1211	0.704	0.732	1128	0.601	0.645	1379
SM + VP	0.700	0.739	1445	0.731	0.785	1379	0.622	0.656	862
II + SCSM	0.688	0.741	844	0.718	0.772	786	0.612	0.655	894
SM + SCSM	0.700	0.728	521	0.724	0.758	498	0.631	0.684	567
II + BOW + RANSAC	0.741	0.770	1511	0.777	0.802	1456	0.687	0.731	1872
SM + BOW + RANSAC	0.757	0.798	1236	0.778	0.822	1231	0.698	0.748	1560
PPV	0.702	0.731	144	0.741	0.761	137	0.635	0.655	168
PPV + Smoothing	0.737	0.768	166	0.758	0.792	156	0.653	0.686	195
PPV + GC	0.761	0.788	150	0.785	0.812	134	0.676	0.710	174
PPV + Smoothing + GC	0.769	0.803	175	0.790	0.830	165	0.685	0.723	202

consequence of reducing the number of stored triplets, the computational time decreases as a much smaller set of triplets must be searched across. The difference between an allocation of 16MB and 8MB was a decrease in computational time from 168ms to 137ms. The competing methods naturally increase in performance as greater memory is allocated, but the scalability of our method is comparable due to the allocation of memory based on an entropy measure, rather than the naive heuristics available for the competitors.

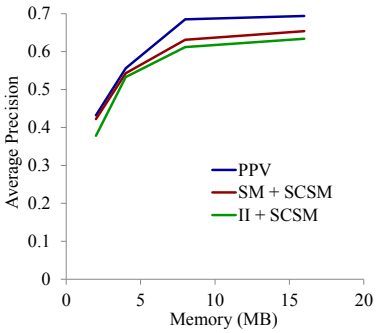


Fig. 6. The effect of the allocated memory on the average precision of recognition

8 Conclusions

In this paper a new framework for fast place recognition has been presented, called Pairwise Probabilistic Voting. It has been shown that it is possible to combine the merits of geometric constraints and inverted-index approaches, by voting for scenes through simple, local pairwise relationships. Geometry can be embedded in the inverted index by discretising image space over a number of geometry types, which also enables a strong generative model to be built, with joint distributions over pairwise appearance and geometry. We have also shown how global geometric constraints can be applied again by simply considering pairwise geometries, offering similar recognition performance to RANSAC approaches at a fraction of the required time. Our PPV method is also able to scale well with modest memory requirements due to its ability to remove most pairwise relationships from the index based on an entropy measure.

References

1. Arandjelovic, R., Zisserman, A.: Three things everyone should know to improve object retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2911–2918 (2012)
2. Cummins, M., Newman, P.: FAB-MAP: Probabilistic localization and mapping in the space of appearance. International Journal of Robotics Research 27, 647–661 (2008)

3. Heath, K., Gelfand, N., Ovsjanikov, M., Aanjaneya, M., Guibas, L.J.: Image-webs: Computing and exploiting connectivity in image collections. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, pp. 3432–3439 (2010)
4. Jegou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 304–317. Springer, Heidelberg (2008)
5. Jégou, H., Chum, O.: Negative evidences and co-occurrences in image retrieval: The benefit of PCA and whitening. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part II. LNCS, vol. 7573, pp. 774–787. Springer, Heidelberg (2012)
6. Johns, E., Yang, G.Z.: From images to scenes: Compressing an image cluster into a single scene model for place recognition. In: Proceedings of the IEEE International Conference on Comptuer Vision, pp. 874–881 (2011)
7. Johns, E., Yang, G.Z.: Generative methods for long-term place recognition in dynamic scenes. pp. 297–314 (2014)
8. Kalantidis, Y., Tolias, G., Avrithis, Y., Phinikettos, M., Spyrou, E., Mylonas, P., Kollias, S.: VIRaL: Visual image retrieval and localization. Multimedia Tools and Applications 51, 555–591 (2011)
9. Li, Y., Crandall, D.J., Huttenlocher, D.P.: Landmark classification in large-scale image collections. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1957–1964 (2009)
10. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal on Computer Vision 60, 91–111 (2004)
11. Mikulík, A., Perdoch, M., Chum, O., Matas, J.: Learning a fine vocabulary. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part III. LNCS, vol. 6313, pp. 1–14. Springer, Heidelberg (2010)
12. Östergård, P.R.: A fast algorithm for the maximum clique problem. Discrete Appl. Math. 120, 197–201 (2002)
13. Philbin, J., Chum, O., Isard, M.: J. Sivic, A.Z.: Lost in quantization: Improving particular object retrieval in large scale image databases. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2008)
14. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
15. Raguram, R., Chum, O., Pollejeys, M., Matas, J., Frahm, J.M.: Usac: A universal framework for random sample consensus. Pattern Analysis and Machine Intelligence 35, 2022–2038 (2013)
16. Raguram, R., Wu, C., Frahm, J.M., Lazebnik, S.: Modeling and recognition of landmark image collections using iconic scene graphs. International Journal of Computer Vision 95, 213–231 (2011)
17. Sattler, T., Leibe, B., Kobbelt, L.: Fast image-based localization using direct 2D-to-3D matching. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 667–674 (2011)
18. Shen, X., Lin, Z., Brandt, J., Wu, Y.: Spatially-constrained similarity measure for large-scale object retrieval. Pattern Analysis and Machine Intelligence 36, 1229–1241 (2014)
19. Tolias, G., Kalantidis, Y., Avrithis, Y., Kollias, S.: Towards large-scale geometry indexing by feature selection. Computer Vision and Image Understanding 120(3), 31–45 (2014)

20. Tolias, G., Avrithis, Y.: Speeded-up, relaxed spatial matching. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1653–1660 (2011)
21. Wang, R., Tang, Z., Cao, Q.: An efficient approximation algorithm for finding a maximum clique using hopfield network learning. Neural Computing 15(7), 1605–1619 (2003)
22. Wang, X., Yang, M., Cour, T., Zhu, S., Yu, K., Han, T.X.: Contextual weighting for vocabulary tree based image retrieval. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 209–216 (2011)
23. Wu, Z., Ke, Q., Isard, M., Sun, J.: Bundling features for large scale partial-duplicate web image search. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, pp. 25–32 (2010)
24. Yuan, U., Wu, Y., Yang, M.: Discovery of collocation patterns: from visual words to visual phrases. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
25. Zhang, Y., Jia, Z., Chen, T.: Image retrieval with geometry-preserving visual phrases. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 809–816 (2011)
26. Zheng, Y., Zhao, M., Song, Y., Adam, H., Buddemeier, U., Bissacco, A., Brucher, F., Chua, T.S., Neven, H.: Tour the world: Building a web-scale landmark recognition engine. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1085–1092 (2009)