Large-Scale Location Recognition and the Geometric Burstiness Problem

Torsten Sattler[1], Michal Havlena[2], Konrad Schindler[3], Marc Pollefeys[1]
[1]Department of Computer Science, ETH Zürich, Switzerland
[2]Computer Vision Laboratory, ETH Zürich, Switzerland
[3]Institute of Geodesy and Photogrammetry, ETH Zürich, Switzerland
{sattlert,pomarc}@inf.ethz.ch, havlena@vision.ee.ethz.ch, schindler@geod.baug.ethz.ch

Abstract

Visual location recognition is the task of determining the place depicted in a query image from a given database of geo-tagged images. Location recognition is often cast as an image retrieval problem and recent research has almost exclusively focused on improving the chance that a relevant database image is ranked high enough after retrieval. The implicit assumption is that the number of inliers found by spatial verification can be used to distinguish between a related and an unrelated database photo with high precision. In this paper, we show that this assumption does not hold for large datasets due to the appearance of geometric bursts*,* i.e.*, sets of visual elements appearing in similar geometric configurations in unrelated database photos. We propose algorithms for detecting and handling geometric bursts. Although conceptually simple, using the proposed weighting schemes dramatically improves the recall that can be achieved when high precision is required compared to the standard re-ranking based on the inlier count. Our approach is easy to implement and can easily be integrated into existing location recognition systems.*

1. Introduction

Given a database of geo-tagged images, the task of a visual location recognition system is to determine the place depicted in a query photo [2, 5, 30, 34, 35]. Knowing which database images show the same place as the query, the position (potentially also the orientation) from which the query photo was taken can either be approximated [7, 13, 37] or computed precisely [26,40] from the known positions of the matching database images. Location recognition techniques play an important role for several applications such as loop-closure in robotics [9], landmark recognition [4, 7], visual navigation [26], and image-based localization [6, 14, 29].

Typically, two tasks must be accomplished to solve the visual location recognition problem: *(i)* find a set of database images visually similar to the query and *(ii)* determine which, if any, of the retrieved images depict the same place as the query. The first step is another canonical problem of computer vision, namely image retrieval [31]. Consequently, most work on location recognition focuses on optimizing the retrieval step, with the aim to maximize the portion of queries where at least one relevant database photo is contained in the N most similar retrieved images, the so-called *recall@N*. Image retrieval largely ignores the spatial relations between features in the query image. Thus, the recall@N can be improved further through *spatial verification*: the (approximate) geometric transformation between the query and the top-ranked images after retrieval is estimated [24,32,33] and the database images are re-ranked based on the number of inliers to the transformation.

Improving the retrieval step is clearly a key factor for solving the location recognition problem. Yet, only improving the recall@N is not sufficient for quite a few applications which require high precision. For example, to support loop-closure in SLAM systems one must recognize previously visited locations with high precision, since the loop closure (a.k.a. pose graph optimization) itself tolerates only a small number of mistakes [18]. Similarly, tools which automatically annotate photos with the place where they were taken [11] become useless if the user must search and correct too many mistakes. Thus, a second key capability of a location recognition system is to decide with high precision (*i.e.*, low false-positive rate) which of the retrieved images actually depict the same place – ideally the bulk of the queries should satisfy recall@1.

As explained, the standard way to refine the raw list of retrieved images is geometric verification with a suitable transformation. But the re-ranking is surprisingly primitive: typically, images are simply re-ordered by the number of inliers to the transformation, respectively discarded if that number falls below some threshold. Interestingly, the results reported in previous work actually suggest that this is not a suitable strategy if high precision is required. *E.g.*, [2] reports a recall@1 of ≈70% on the large-scale Pittsburgh dataset [35], but a recall@50 of ≈90%. In other words, for

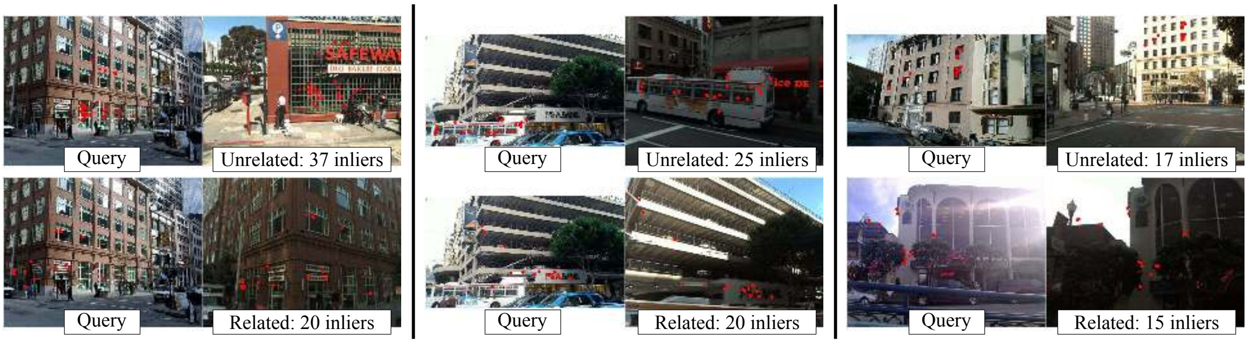

Figure 1. Geometric bursts, *i.e.*, geometrically consistent structures of similar appearance, cause problems to location recognition: database images depicting an unrelated place can often attain more inliers to the estimated geometric model than photos of the same place.

20% of all query images an unrelated image has a higher inlier count than any photo taken at the same place, even after spatial verification! Somewhat surprisingly, this observation has received very little attention in the literature.

It is known that visual words appearing in *visual bursts* [16] or words that are likely to co-occur together [8] require special handling. In much the same way, large databases often contain *geometric bursts*, *i.e.*, geometric configurations of visually similar features that are shared between different places. Fig. 1 illustrates this phenomenon. By definition, *geometric bursts* appear in multiple locations of the database. Hence, they violate the basic assumption underlying geometric verification: even spatial configurations of several features are not always unique, making it impossible to distinguish between a correct and a wrong location.

In this paper, we investigate ways to explicitly handle geometric bursts by analyzing the geometric relations between the different database images retrieved by a query. Namely, we make the following contributions: *(i)* we introduce the concept of geometric burstiness. *(ii)* we demonstrate that geometric bursts are an important cause for false positives and have a significant impact on the precision of location recognition. *(iii)* we show how to dramatically increase the recall for a given precision with an appropriately weighted inlier count that better accounts for geometric bursts. *(iv)* our approach is designed such that it operates online at query time, and requires neither costly preprocessing nor any additional storage. It can be used as a drop-in replacement for the conventional inlier count, without any changes to the underlying retrieval system, and we make source code available at `https://github.com/tsattler/geometric_burstiness`.

2. Related Work

Location recognition, also referred to as *place recognition*, relies on image retrieval techniques such as inverted files [31], quantized feature matching with large visual vocabularies [22, 24], vocabulary trees [23], and fast approximate spatial matching [24, 33]. Hamming embedding [15] simulates the similarity between two descriptors at little additional run-time and memory overhead by using compact binary representations. Thus, Hamming embedding allows to remove many of the unrelated votes caused by visual word quantization [29]. To overcome the limited viewpoint invariance of modern features such as SIFT [20], [4, 7, 26] rectify images prior to feature extraction, with the help of vanishing points. [7] show that combining rectified and regular images increases the overall performance. Instead of using invariant features, [34] densely sample the scene by generating synthetic renderings from novel viewpoints.

Recent work on place recognition focused on the problems caused by repetitive structures and uninformative features. Repetitive structures lead to bursts of visual elements, *i.e.*, a visual word occurs very often in an image [16]. While [16] handle the repetitions of a single word, [8] detect and handle sets of co-occurring features, showing that the classical *tf-idf* weighting cannot handle that case. [35] recognize that repetitive structures are not only a nuisance, but can provide valuable information about a place. They propose to consider the features in a repetitive pattern as a soft assignment of a single visual element, and show that their explicit handling of repetitions outperforms the standard scheme [16] that down-weights visual bursts.

In order to improve and accelerate the retrieval performance, [30, 36] select only an informative subset of all database features that are repeatable and/or unique for each place. [17] proceed more conservatively and only remove confusing features that are also found in unrelated places of the database. Both [6, 12] learn SVM classifiers on top of the Bag-of-Words image representation for each place, so as to properly weight informative and confusing features. All these methods [6, 12, 17, 30, 36] must query every single database photo against the database. [2] argue that this is infeasible for large databases due to its quadratic computational complexity. Instead, they propose to handle repetitions and uninformative features online at query time, by density estimation in the space of Hamming descriptors, which can be computed efficiently. All these methods aim to improve the retrieval stage *before* spatial verification. In contrast, we focus on providing a better measure for deciding between related and unrelated places *after* verification.

Instead of using visual words, [37, 38] exploit the full feature descriptors for matching. They do not vote for individual database images, but instead use the geo-tags of all matching images to cast votes for the geo-position of the query image. [39] take that idea one step further and use a 3D model of the scene to better constrain the voting, for both the position and orientation of the image. However, using full descriptors soon becomes infeasible at large scale.

Closely related to location recognition is the image-based localization problem, where the goal is to recover the full camera pose of a given query image relative to a 3D scene model [19, 28]. Image-based localization systems put emphasis on computing the camera pose with a high precision. They use the full feature descriptors for matching and more restrictive geometric models for spatial verification, while image retrieval-based approaches traditionally use visual words for matching and approximate geometric models for verification. As a result, image-based localization by large achieves a much higher recall than place recognition methods in the high precision regime, although [27] recently showed that a similar recall at high precision can be achieved with quantized features as well. However, there are no theoretical reasons why location recognition approaches should perform worse. In this paper, we show that by handling geometric bursts, location recognition approaches can reach similar or better levels of recall in the high precision regime without using the full descriptors and using only an affine model for geometric verification.

3. Geometric Burstiness

During conventional image retrieval the spatial configuration of features in query and database images is ignored. As a consequence, a retrieved database image may contain many visually similar features, but in a very different geometric configuration. The purpose of geometric verification is to detect images where the feature point locations are not consistent, *i.e.*, they are unrelated and retrieved by mistake. The common assumption is that, if one fits a suitable image-to-image transformation to the feature matches, not many inliers will be found for unrelated images. The inlier count is used to re-rank the top-k retrieved images. Clearly, the assumption does not hold if the same geometric configuration occurs repeatedly in the database. Such non-unique configurations, *geometric bursts*, are more likely if the scene is large, and if it contains visually similar objects. The central message of this paper is that, other than what one might hope, geometric bursts do occur regularly in realistic databases. That means that one will encounter cases where unrelated images have the highest inlier counts (*c.f.* Fig. 1(left & middle)). With standard re-ranking these attain the highest rank and lower the recall@N. Note that geometric bursts are not restricted to small image areas.

However, the impact of geometric bursts goes beyond a reduced recall@N: as unrelated database images can have many inliers for some query images, the inlier count is also not a suitable measure to decide whether a place has been correctly recognized or not (*c.f.* Fig. 1(right)). Fig. 2 shows that this can result in a much lower recall in the high precision regime, when one must set a high threshold, *e.g.*, 90%, to the amount of the correct answers returned by the system.

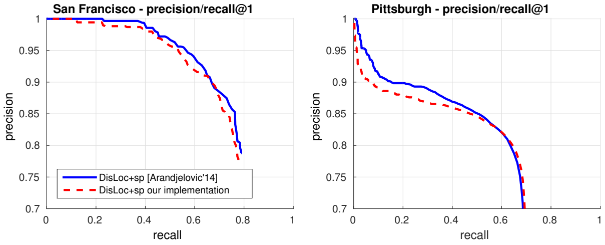

Figure 2. Precision-recall curves obtained by a state-of-the-art localization recognition system [2] when considering the top-ranked image after spatial verification (raw inlier count re-ranking).

One obvious strategy to handle geometric bursts is to remove them in a pre-processing step, by detecting visual bursts for each database image, similar to the removal of confusing features proposed in [17]. However, the computational complexity of such an offline process is quadratic in the number of database images: each image needs to be queried against the complete database. As pointed out by [2], such preprocessing quickly becomes infeasible as the database size grows. We propose to instead handle geometric bursts at query time. In [8] it has been shown how to efficiently detect *co-occurrence sets*, *i.e.*, sets of visual words likely to appear together, in the query image during retrieval. However, it is unclear how to distinguish co-occurrence sets between multiple images of the *same* location from geometric bursts that appear at unrelated locations. Moreover, removing bursts at retrieval time runs the risk of also losing the correct location [8]. We thus prefer to handle geometric bursts at the stage where they cause problems, *i.e.*, *after* spatial verification. By definition, geometric bursts visible in a query image will appear in multiple unrelated database images. Given the geo-tags of the database photos and the inlier matches detected for them, it is therefore rather simple to detect geometric bursts on demand and down-weight their influence on the image ranking. We will show in Sec. 5 that it is easier to distinguish related and unrelated images with that weighted inlier count. As a result, our approach greatly increases the recall at high precision.

4. Detecting and Handling Geometric Bursts

Essentially, a *geometric burst* is a set of visual words that co-occur repeatedly in the same spatial configuration. [16] show for *visual bursts* that appropriate down-weighting improves retrieval, and [8] apply the same weighting scheme for co-occurrence sets (sets of co-occurring features in an arbitrary spatial configuration). In Sec. 4.1, we first review this weighting scheme and discuss how to adapt it for geometric burstiness. While this simple adaptation already improves the recall at high precision, it overestimates the importance of geometric bursts. Sec. 4.2 describes how to remedy this behavior. Sec. 4.3 then proposes to measure place popularity and include it into the weighting scheme.

4.1. Inter-Image Geometric Burstiness

According to [16], a visual burst is a visual word which violates the assumption that words appear independently of each other. Moreover, they distinguish between *intra-image* and *inter-image* burstiness. Intra-image bursts are caused by repetitive structures found in a single image, whereas inter-image burstiness refers to visual elements shared between many database images. In terms of geometric burstiness, intra-image bursts can easily be handled by enforcing one-to-one correspondences for the inliers.

[16] handle inter-image visual burstiness as follows: let $\text{sim}(\mathcal{Q}_i, \mathcal{D}^m_j)$ be the similarity score between the i^{th} feature in the query image $\mathcal{Q}$ and the j^{th} feature in the m^{th} database image $\mathcal{D}^m$, *e.g.*, computed via Hamming embedding. The sum of similarity scores for the i^{th} query feature across all database images is thus given by

$$\text{sim}_{\sum}(\mathcal{Q}_i) = \sum_m \sum_j \text{sim}(\mathcal{Q}_i, \mathcal{D}^m_j) \ . \tag{1}$$

[16] use this sum to weight each similarity score $\text{sim}(\mathcal{Q}_i, \mathcal{D}^m_j)$ by multiplying with $\sqrt{\frac{\text{sim}(\mathcal{Q}_i, \mathcal{D}^m_j)}{\text{sim}_{\sum}(\mathcal{Q}_i)}}$.

Obviously, this weighting scheme can be adapted to the case of geometric bursts. A match $(\mathcal{Q}_i, \mathcal{D}^m_j)$ contributes a value of 1 to the inlier count for image $\mathcal{D}^m$ if it is an inlier to the estimated model and a value of 0 otherwise, *i.e.*,

$$\text{sim}_{\text{geo}}(\mathcal{Q}_i, \mathcal{D}^m_j) = \begin{cases} 1 & \text{if } (\mathcal{Q}_i, \mathcal{D}^m_j) \text{ is an inlier} \\ 0 & \text{otherwise} \end{cases} \ . \tag{2}$$

The i^{th} query feature $\mathcal{Q}_i$ is a part of at most one inlier match for any database image, thus the geometric equivalent

$$\text{sim}_{\text{geo},\sum}(\mathcal{Q}_i) = \sum_{m=1}^{k} \sum_j \text{sim}_{\text{geo}}(\mathcal{Q}_i, \mathcal{D}^m_j) \tag{3}$$

to Eqn. (1) is simply the number of database images for which $\mathcal{Q}_i$ is an inlier of the geometric verification. Notice that while Eqn. (1) considers all database images, Eqn. (3) only includes the top-k ranked images after retrieval for which spatial verification is performed[1].

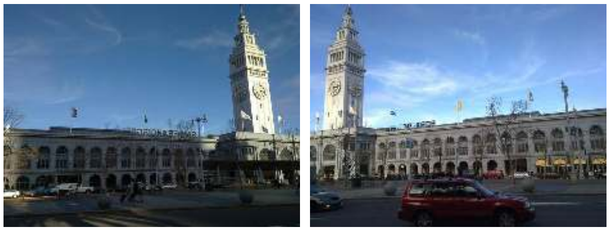

Figure 3. Two images from the San Francisco dataset depicting the same clock tower from different viewpoints.

A query feature $\mathcal{Q}_i$ participates in a geometric burst if it forms part of the inlier set for at least two database images $\mathcal{D}^m \neq \mathcal{D}^l$. To assign a lower weight to features from a geometric burst, we use an *inter-image-weighted* inlier count

$$\text{I}_{\text{inter-image}}(\mathcal{D}^m) = \sum_{\text{inlier match } (\mathcal{Q}_i, \mathcal{D}^m_j)} \frac{1}{\sqrt{\text{sim}_{\text{geo},\sum}(\mathcal{Q}_i)}} \tag{4}$$

over all verified matches $(\mathcal{Q}_i, \mathcal{D}^m_j)$ from the query image $\mathcal{Q}$ to the database image $\mathcal{D}^m$. In Sec. 5, we will experiment with various weighting functions, as well as a variant that completely removes features from geometric bursts.

4.2. Inter-Place Geometric Burstiness

Usually, place recognition databases contain multiple photos of each place, *e.g.*, street-level panoramas taken at regular intervals as in online mapping services. Two images depicting the same location will inherently share common features. Thus, it is likely that a geometric burst detected within one of them is also detected in the other one. Eqn. (3) treats each view separately and thus overestimates the burstiness of the underlying features. Consequently, the weighted inlier count, Eqn. (4), underestimates the similarity between the query and database images. Rather than identifying geometric bursts on a per-image level, it would be more appropriate to identify bursts on a per-place level. In the following, we provide a workable definition of a "place" and with that definition compute an *inter-place* burstiness measure.

Defining places. Fig. 3 shows a fundamental difficulty of visual location recognition. Two database images were taken at different places that are far apart, but they contain the same clock tower. Note the subtle problem: features on the front side of the tower in the two images depict the same physical points, nevertheless they form a geometric burst, since the tower is visible from multiple locations and can confuse place recognition. The example highlights a simple, but important fact: visual similarity alone is unsuitable to define a place (even if one had perfect descriptors that unambiguously encode 3D points), simply because vision is a long-range sensor. To solve this problem, we must exploit the geo-tags $g(\mathcal{D}^m) \in \mathbb{R}^2$ of each database image, obtained, *e.g.*, from GPS or Structure-from-Motion.

[1] To avoid confusion, top-k will refer to the k images with the highest similarity score *after retrieval*. The recall@N measure then considers the N highest ranked images after applying spatial verification on the top-k images and re-ranking based on the (raw or weighted) number of inliers.

One natural approach is to define a place as the set of all database images whose geo-tags fall into a pre-defined cell in scene space, *e.g.*, on a regular lattice or a Voronoi tessellation found with k-means clustering. A regular grid is tempting, but will lead to quantization artifacts near the cell boundaries, because it ignores the distribution of the geo-tags. We therefore adaptively cluster at query time based on the spatially verified database images. At first glance, this approach seems suboptimal. However, we only need to consider a few spatially verified images[2]. Compared to the time required for retrieval and spatial verification, we found the time required for clustering to be negligible.

Our method is inspired by the initialization procedure of k-means++ clustering [3]: $\mathcal{D}^1$ is the database image with the largest number of inliers. Its geo-tag $g(\mathcal{D}^1)$ defines the center of the first cluster. We iteratively select the database image $\mathcal{D}^m$ furthest away from all previously chosen cluster centers. This process is terminated once there exists no more image $\mathcal{D}^m$ that is more than d_{max} meters away from its closest cluster center, or all k verified images have been considered. The termination criterion is chosen to reflect that nearby images should belong to the same place. Next, we assign each verified database image $\mathcal{D}^m$ to its closest cluster center $c(\mathcal{D}^m)$. Each cluster then defines one place.

Inter-place burstiness. Given the set of places obtained via clustering, we adapt the geometric burstiness weighting scheme to avoid overestimating the number of geometric bursts. For a feature $\mathcal{Q}_i$ in the query image, let $\mathcal{D}(\mathcal{Q}_i)$ be the set of database images containing an inlier match for $\mathcal{Q}_i$. The set of relevant places is then given by

$$c(\mathcal{Q}_i) = \{c(\mathcal{D}^m) \mid \mathcal{D}^m \in \mathcal{D}(\mathcal{Q}_i)\} \ . \tag{5}$$

We can now normalize over places rather than images to define an *inter-place-weighted* inlier count

$$\text{I}_{\text{inter-place}}(\mathcal{D}^m) = \sum_{\text{inlier match}(\mathcal{Q}_i, \mathcal{D}^m_j)} \frac{1}{|c(\mathcal{Q}_i)|} \ . \tag{6}$$

Compared to Eqn. (4), Eqn. (6) counts each geometric burst at most once per place to assess a query feature $\mathcal{Q}_i$. In the experiments, we show that this greatly improves recall at high precision. In Eqn. (6), we have dropped the square root as we found that the new criterion performs slightly better without it. We will experiment with different weighting functions in Sec. 5.

Exploiting metadata. Some datasets provide detailed metadata for each database image. For example, the San Francisco dataset [7] provides a "carto id", a unique identifier for the building visible in each database image. Naturally, this information can be used as an alternative way of defining places. In Sec. 5 we show that using such metadata does not necessarily improve over the data-driven clustering, possibly because the "carto id" is somewhat ambiguous if more than one building is visible in the foreground.

4.3. Inter-Place Burstiness with Place Popularity

So far, we proposed a method which weights the individual features of $\mathcal{Q}$ differently in the computed inlier sum. Once we have the database images clustered to places, we can also use the *popularity* of the individual places to further refine the weighting scheme. For a database image $\mathcal{D}^m$, let $\mathcal{C}(\mathcal{D}^m)$ be the set of images from its place. The place's popularity $p(\mathcal{C}(\mathcal{D}^m))$ is given as the number of features from $\mathcal{Q}$ which are inliers for at least one of the images in $\mathcal{C}(\mathcal{D}^m)$:

$$p(\mathcal{C}(\mathcal{D}^m)) = |\{i \mid \mathcal{D}(\mathcal{Q}_i) \cap \mathcal{C}(\mathcal{D}^m) \neq \emptyset\}| \ . \tag{7}$$

The *inter-place-popularity-weighted* inlier count is then defined in the following way:

$$\text{I}_{\text{inter-place + pop}}(\mathcal{D}^m) = \text{I}_{\text{inter-place}}(\mathcal{D}^m) \cdot \frac{p(\mathcal{C}(\mathcal{D}^m))}{\max\limits_{l} p(\mathcal{C}(\mathcal{D}^l))} . \tag{8}$$

Therefore, all retrieved database images not located at the most popular place are further down-weighted.

4.4. Discussion

The weighting scheme proposed above is conceptually simple, and very easy to implement. It requires neither any additional matching or verification steps nor any external data. Notwithstanding its simplicity, re-weighting according to geometric burstiness brings drastic improvements compared to the traditional inlier count, as we will show in Sec. 5. The simplicity of our method naturally raises the question whether a different, possibly more sophisticated, way of handling bursts would perform even better.

In Sec. 5.2, we experiment with different weighting schemes for both Eqn. (4) and Eqn. (6). For example, we use $\sqrt{|c(\mathcal{Q}_i)|}$ instead of $|c(\mathcal{Q}_i)|$ in Eqn. (6), to assign more importance to inliers found on geometric bursts. Our results will show that changing the weighting function has only a small impact on the overall performance of Eqn. (6), which is in agreement with the results of [16] for *visual* bursts. At the same time, we observe a significant loss when Eqn. (4) is used instead. This suggests that detecting which geometric bursts come from the same scene structure is more important than the exact weighting function. We also show that the performance of the proposed method depends only little on the exact definition of what constitutes a place.

There is one obvious difference between visual and geometric bursts: visual bursts (and similarly co-occurrence sets) are defined independent of the feature's position in the image. In contrast, geometric bursts essentially correspond to geometrically consistent regions in the images. We tried to account for this difference by dividing the query image

[2] Typically, 10 to 1000 top-ranked images are spatially verified.

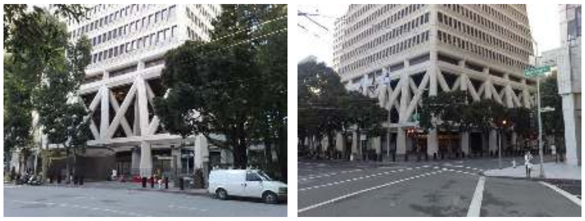

Figure 4. Two images from the San Francisco dataset prominently displaying the same building from different sides.

into tiles and counting the number of geometric bursts per tile rather than per feature, but this did not improve the results (improper setting of tile size even worsens the results).

There is one obvious situation in which the weighting scheme fails: consider the building in Fig. 4. If all inliers are found on a surface visible from many places, down-weighting them will have no effect on the ranking. In such cases, higher-level information is needed, *e.g.*, reasoning based on the outlines of the nearby buildings.

5. Experimental Evaluation

In this section, we evaluate the weighting schemes for geometric bursts proposed in Sec. 4 on two standard benchmark datasets for place recognition, San Francisco [7] and Pittsburgh [35]. We show that accounting for geometric bursts significantly improves the recall in the high precision regime as well as the overall recall.

San Francisco Landmarks dataset [7]. The San Francisco dataset consists of 1.06M database images extracted from about 150k panoramic images captured by a vehicle driving through the streets of San Francisco. The 803 query images are taken with multiple mobile phones. Each database image is annotated with a "carto id" denoting the building visible in the image and a list of relevant "carto ids" is also provided for each query image. A query image is then considered to be successfully localized if the top-ranked database image is annotated with a relevant "carto id". We use the 2014 version of the ground truth [2].

Pittsburgh dataset [35]. The database photos for the Pittsburgh dataset were obtained by extracting 254k perspective images from about 10.6k panoramas downloaded from Google Street View (which leads to a rather large distance between the panorama locations). 24k query images then come from a separate set of Google Street View panoramas taken from Google's Pittsburgh Research Dataset. Both sets of panoramas have quite accurate GPS coordinates which defines the localization task: A query image is considered being localized if the GPS position of the top-ranked database photo is within 25m of the query image's position.

Place recognition pipeline [2]. We use our own implementation of a state-of-the-art location recognition system [2], referred to as *DisLoc*, to perform image retrieval and spatial verification. DisLoc uses 64-bit Hamming embedding [15] to compute the similarity between a query and database feature, which is weighted based on the density of the descriptor space surrounding the database feature. As a result, less weight is assigned to matches found in dense parts of the descriptor space, effectively down-weighting visual elements that appear often. Inter-image *visual* burstiness weighting [16] is used to handle visual burst. As in [2], upright RootSIFT [1, 20] descriptors are extracted from Hessian-Affine keypoints [21] and are assigned to the closest out of 200k words. To lessen quantization artifacts, each query feature is assigned to its 5 closest words. As in [2], fast approximate spatial verification with an affine model [24] is used to verify the top-200 images found by the retrieval step. Fig. 2 shows that our implementation performs slightly worse than [2], *i.e.*, the improvements reported in this paper do not come from a better implementation.

5.1. Baseline Comparisons

First, we compare the weighting schemes for geometric bursts proposed in Sec. 4 with two baselines: the *raw* inlier count and the *effective* inlier count [14, 27]. The latter measure is defined as follows: each inlier feature in the query image covers the area A_i contained in a circle of radius r around itself, with r set to 12 pixels in our experiments. Given n inliers, the effective inlier count is computed as

$$\mathrm{I}_{\text{eff}} = \frac{|\bigcup_i A_i|}{\sum_{i=1}^{n} |A_i|} \cdot n \ , \tag{9}$$

where $|A_i| = \pi \cdot r^2$ denotes the size of the area covered by the i^{th} inlier feature. Eqn. (9) thus compares the actual area covered by all inliers with the area that can be covered if none of the circles are overlapping. This measure down-weights inliers found in a small region of the query image. Following the setup from [7], we obtain precision-recall curves by varying a threshold on the number of inliers for the *raw* count and thresholds on the weighted inlier counts for the *effective* and the two burstiness inlier counts.

The goal of our first experiment is to show that both the *inter-image-weighted* inlier count $\mathrm{I}_{\text{inter-image}}$, Eqn. (4), and the *inter-place-weighted* inlier count $\mathrm{I}_{\text{inter-place}}$, Eqn. (6), enable us to find a better threshold to distinguish between correct and wrong place recognitions. Thus, in this experiment, we only consider the verified database images with the largest *raw* number of inliers found for each query image and do not re-rank based on the weighted inlier counts. As can be seen in Fig. 5(a-b), the *effective* inlier count consistently outperforms the *raw* inlier count, while in turn both burstiness measures outperform the *effective* inlier count. The latter shows that geometric bursts are not restricted to small image regions.

The improvement in recall we gain by accounting for geometric bursts is dramatic: At 95% precision, the *raw* and

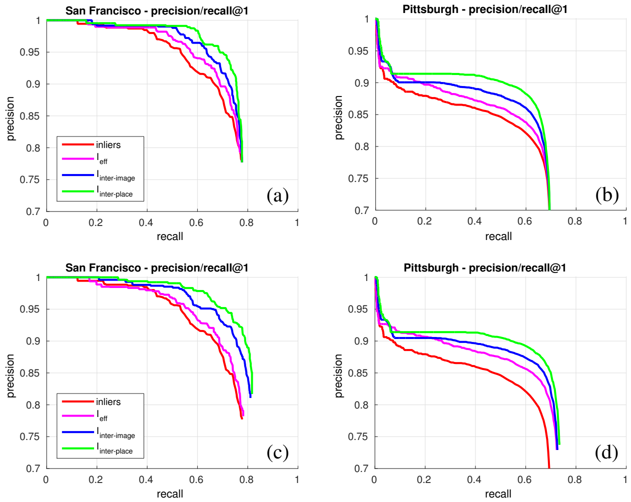

Figure 5. Comparison against baseline measures: *(a-b)* without and *(c-d)* with re-ranking using the respective measure.

effective inlier counts achieve 53.4% respectively 57.3% recall on San Francisco. In contrast, our weighting schemes for geometric burstiness achieve 63.5% and 70.1%. At 90% precision, the *raw* and *effective* counts obtain 7.2% and 18% recall on Pittsburgh while the *inter-image* count obtains 25.8% and the *inter-place* measure achieves 51.1% recall. These results clearly demonstrate the importance of handling geometric bursts. Interestingly, all measures perform poorly on the Pittsburgh dataset when a precision higher than 90% is required. We visually inspected over 400 out of 6381 query images for which the top-ranked database photo is unrelated but still receives a high weighted inlier score. One common failure case is that all inliers are solely found on geometric bursts, *e.g.*, identical facades of a building or buildings seen from afar. As discussed in Sec. 4.4, such cases cannot be resolved by considering geometric bursts.

Fig. 5(c-d) demonstrates that accounting for geometric bursts not only enables a better decision between correctly and incorrectly retrieved places. It also improves the overall recall when used for re-ranking.

5.2. Ablation Study

In the next experiment, we evaluate the impact of different parameter settings for the burstiness schemes.

Different weighting schemes. We test the impact of different weighting schemes for the number of geometric bursts. For example, we replace the term $1/\sqrt{\text{sim}_{\text{geo},\sum}(\mathcal{Q}_i)}$ in Eqn. (4) with $1/\text{sim}_{\text{geo},\sum}(\mathcal{Q}_i)$ to give less weight to inliers participating in many geometric bursts. In addition, we experiment with removing inliers participating in geometric bursts, *i.e.*, inliers to two or more images, respectively places. Fig. 6(a-b) shows the results from this ablation study. As can be seen, the weighting function used has a large impact on the *inter-image* count since, *e.g.*, $\text{sim}_{\text{geo},\sum}(\mathcal{Q}_i)$ overestimates the number of geometric bursts for each query feature. Removing inliers participating in bursts further decreases the performance as it does not account for the fact that multiple database photos can depict the same place. The fact that the linear weighting performs similar as the square-root weighting on Pittsburgh comes from the fact that the database images are taken further apart, so there are fewer photos depicting the same place.

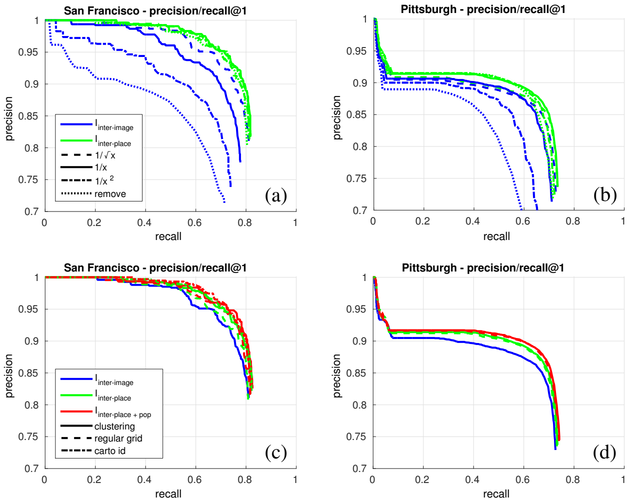

Figure 6. Ablation study for our method on the two datasets. All results are after re-ranking using the respective measure. Colors denote different inlier counts, while the line style, *e.g.*, dashed or dotted, denotes different ways to compute these counts.

In contrast to the *inter-image* count, the *inter-place* count is much less sensitive to the weighting function used, with the linear weighting performing slightly better than the other weighting functions. This demonstrates that the main importance lies in detecting related bursts rather than in the way bursty inlier features are weighted.

Different place definitions. So far, we have only used the place clustering scheme described in Sec. 4.2, with the maximum distance set to $d_{\max} = 25$m. Next, we compare this scheme against using a regular grid of side length 25m. For San Francisco, we also compare against using the "carto id" of the database images for clustering. For the *inter-place* count, we use the linear weighting. The results of the experiments are shown in Fig. 6(c-d). On San Francisco, where the database images are taken more densely, the adaptive clustering scheme performs better than the fixed grid, offering a recall of 71.2% at 95% precision compared to 65.6% for the fixed grid. However, using the "carto ids" to define places does not offer a significant advantage.

The sparser sampling on the Pittsburgh dataset leads to less quantization artifacts. As a result, using either the adaptive clustering or the regular grid results in virtually the same recall-precision curve. Independently of the place definition, the *inter-place* count gives better results than the *inter-image* count. This again demonstrates the importance of accounting for the fact that multiple database images can depict the same part of the scene.

Popularity-based weighting. Fig. 6(c-d) show the results obtained with the *inter-place-popularity-weighted* inlier count (using the place clustering scheme described in Sec. 4.2). On the San Francisco dataset, the improvement compared to the *inter-place* count is modest as the recall at 95% precision increases from 71.2% to 72.4%. However, the improvement measured on Pittsburgh is more significant as the recall at 90% precision increases from 54.3% to 59.4%. The *inter-place-popularity* count penalizes database images that do not come from the place with the largest number of inliers. The smaller improvement on San Francisco can be explained by the fact that location recognition performs better on this dataset, *i.e.*, most of the correctly retrieved images come from the most popular place.

The maximum recall our implementation can achieve when verifying the 200 top-ranked images is 87.92% for San Francisco and 88.98% for Pittsburgh, respectively. Using the *inter-place-popularity* count, we achieve a recall@1 of 82.57% for San Francisco and 74.15% for Pittsburgh.

5.3. Comparison with State-of-the-Art

We compare our implementation of *DisLoc+inter-place-popularity* with state-of-the-art place recognition [2, 7, 35] and image-based localization approaches [27, 39]. Since there is no 3D model for the Pittsburgh dataset, the comparison is only performed on San Francisco. [27, 39] use a 3D model provided by [19] while all other methods only use images. Whereas our approach considers the relationship between inliers in multiple images, all other methods score images and/or poses independently of each other. For [7,27,35], we use results kindly provided by the authors to draw the precision-recall curves.

The *Adaptive weights* method from [35] does not perform spatial verification. Thus, we use the similarity scores after retrieval to obtain the precision-recall curve. [7] use histogram equalization before extracting upright SIFT features [20] and a GPS prior (*Hist.Eq. w/ GPS*). The method from [39] uses a 3D model to vote for the most likely camera pose, followed by a RANSAC-based refinement step [10]. The *Hyperpoints* approach from [27] uses a fine vocabulary of 16M words [22] instead of the original point descriptors to obtain the 2D-3D matches required for pose estimation. For completeness, we also report the results originally obtained by [2] with spatial verification (*DisLoc+sp*).

Fig. 7 shows the results of the comparison. As can be seen, [2] can significantly outperform existing methods simply by accounting for geometric bursts. Compared to [7], we improve recall from 70.1% to 80.5% for 90% precision. For 95% precision, we improve the 63.5% recall achieved by [27] to 72.4%. Our recall is close to the 74.2% obtained by [39] with the help of a GPS prior and higher than the 67.5% reported by [39] without a GPS prior.

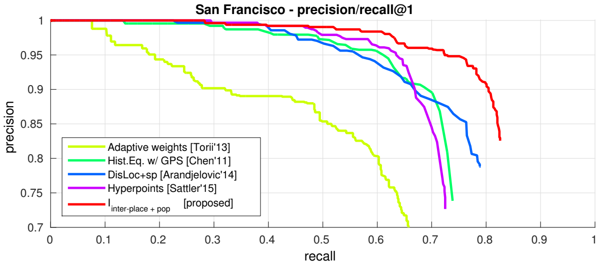

Figure 7. Combining [2] with our proposed scheme for handling geometric bursts not only provides significantly better results over the original method but also outperforms state-of-the-art methods for both place recognition and image-based localization.

	inliers	I_{eff}	$I_{\text{inter-image}}$	$I_{\text{inter-place}}$	$I_{\text{inter-place+pop}}$
Oxford105k (mAP)	0.710	0.730	0.708	0.735	**0.745**
Paris106k (mAP)	0.613	0.619	0.611	0.649	**0.682**

Table 1. Image retrieval results reporting mean average precision.

5.4. Image Retrieval Results

Finally, we show that handling geometric bursts also improves standard image retrieval performance. We trained vocabularies with 200k words on Paris6k [25] and Oxford5k [24] for Oxford105k [24] and Paris106k [25], respectively. Spatial verification is performed for the 1000 top-ranked images. Due to a lack of geo-tags, we define places based on the filename prefixes of the images, *e.g.*, “keble”, which corresponds to the original Flickr queries. Tab. 1 shows the mean average precision (mAP) values obtained with the different (weighted) inlier counts. Both *inter-place* burstiness variants outperform the *raw* and *effective* inlier counts, while the *inter-image* scheme overestimates the number of geometric bursts and performs worse.

6. Conclusions

In this paper, we have shown that *geometric bursts*, *i.e.*, sets of visual elements that appear in a consistent spatial configuration in multiple unrelated database images, can significantly impact the recall that can be achieved by location recognition approaches. We have proposed a simple and easy-to-implement method for detecting and down-weighting geometric bursts. Our approach can serve as a drop-in replacement for the classic re-ranking after spatial verification based on the number of inliers and our experimental results show that this simple approach dramatically increases the recall in the high precision regime. Just by using our weighting scheme, an existing place recognition method achieves state-of-the-art localization performance.

Acknowledgements. This work was supported by Google’s Project Tango and EC Horizon 2020 project REPLICATE (no. 687757). The authors thank Relja Arandjelović for his invaluable help with the DisLoc algorithm.

References

[1] R. Arandjelović and A. Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, 2012.

[2] R. Arandjelović and A. Zisserman. DisLocation: Scalable descriptor distinctiveness for location recognition . In *ACCV*, 2014.

[3] D. Arthur and S. Vassilvitskii. K-means++: The Advantages of Careful Seeding. In *SODA*, 2007.

[4] G. Baatz, K. Köser, D. Chen, R. Grzeszczuk, and M. Pollefeys. Leveraging 3D City Models for Rotation Invariant Place-of-Interest Recognition. *IJCV*, 96(3):315–334, 2012.

[5] G. Baatz, O. Saurer, K. Köser, and M. Pollefeys. Large Scale Visual Geo-Localization of Images in Mountainous Terrain. In *ECCV*, pages 517–530. Springer, 2012.

[6] S. Cao and N. Snavely. Graph-Based Discriminative Learning for Location Recognition. *IJCV*, 112(2):239–254, 2015.

[7] D. Chen, G. Baatz, K. Köser, S. Tsai, R. Vedantham, T. Pylvänäinen, K. Roimela, X. Chen, J. Bach, M. Pollefeys, B. Girod, and R. Grzeszczuk. City-scale landmark identification on mobile devices. In *CVPR*, 2011.

[8] O. Chum and J. Matas. Unsupervised Discovery of Co-occurrence in Sparse High Dimensional Data. In *CVPR*, 2010.

[9] M. Cummins and P. Newman. FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. *IJRR*, 27(6):647–665, 2008.

[10] M. Fischler and R. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Comm. ACM*, 24(6):381–395, 1981.

[11] S. Gammeter, T. Quack, and L. Van Gool. I Know What You Did Last Summer: Object-Level Auto-Annotation of Holiday Snaps. In *ICCV*, 2009.

[12] P. Gronat, G. Obozinski, J. Sivic, and T. Pajdla. Learning per-location classifiers for visual place recognition. In *CVPR*, 2013.

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

[14] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof. From Structure-from-Motion Point Clouds to Fast Location Recognition. In *CVPR*, 2009.

[15] H. Jegou, M. Douze, and C. Schmid. Hamming Embedding and Weak Geometric Consistency for Large Scale Image Search. In *ECCV*, 2008.

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

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

[18] G. H. Lee, F. Fraundorfer, and M. Pollefeys. Robust Pose-Graph Loop-Closures with Expectation-Maximization. In *IROS*, 2013.

[19] Y. Li, N. Snavely, D. P. Huttenlocher, and P. Fua. Worldwide Pose Estimation Using 3D Point Clouds. In *ECCV*, 2012.

[20] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[21] K. Mikolajczyk and C. Schmid. Scale & Affine Invariant Interest Point Detectors. *IJCV*, 60(1):63–86, 2004.

[22] A. Mikulík, M. Perdoch, O. Chum, and J. Matas. Learning vocabularies over a fine quantization. *IJCV*, 103(1):163–175, 2013.

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

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

[25] 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.

[26] D. Robertson and R. Cipolla. An image-based system for urban navigation. In *BMVC*, 2004.

[27] T. Sattler, M. Havlena, F. Radenovic, K. Schindler, and M. Pollefeys. Hyperpoints and Fine Vocabularies for Large-Scale Location Recognition. In *ICCV*, 2015.

[28] T. Sattler, B. Leibe, and L. Kobbelt. Improving Image-Based Localization by Active Correspondence Search. In *ECCV*, 2012.

[29] T. Sattler, T. Weyand, B. Leibe, and L. Kobbelt. Image Retrieval for Image-Based Localization Revisited. In *BMVC*, 2012.

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

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

[32] H. Stewénius, S. H. Gunderson, and J. Pilet. Size Matters: Exhaustive Geometric Verification for Image Retrieval. In *ECCV*, 2012.

[33] G. Tolias and Y. Avrithis. Speeded-up Relaxed Spatial Matching. In *ICCV*, 2011.

[34] A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, and T. Pajdla. 24/7 place recognition by view synthesis. In *CVPR*, 2015.

[35] A. Torii, J. Sivic, T. Pajdla, and M. Okutomi. Visual Place Recognition with Repetitive Structures. In *CVPR*, 2013.

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

[37] A. R. Zamir and M. Shah. Accurate Image Localization Based on Google Maps Street View. In *ECCV*, 2010.

[38] A. R. Zamir and M. Shah. Image Geo-localization Based on Multiple Nearest Neighbor Feature Matching using Generalized Graphs. *PAMI*, 36(8):1546–1558, 2014.

[39] B. Zeisl, T. Sattler, and M. Pollefeys. Camera Pose Voting for Large-Scale Image-Based Localization. In *ICCV*, 2015.

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