PAIGE: PAirwise Image Geometry Encoding for Improved Efficiency in Structure-from-Motion

Johannes L. Schönberger, Alexander C. Berg, Jan-Michael Frahm
Department of Computer Science, The University of North Carolina at Chapel Hill
{jsch,aberg,jmf}@cs.unc.edu

Abstract

Large-scale Structure-from-Motion systems typically spend major computational effort on pairwise image matching and geometric verification in order to discover connected components in large-scale, unordered image collections. In recent years, the research community has spent significant effort on improving the efficiency of this stage. In this paper, we present a comprehensive overview of various state-of-the-art methods, evaluating and analyzing their performance. Based on the insights of this evaluation, we propose a learning-based approach, the PAirwise Image Geometry Encoding (PAIGE), to efficiently identify image pairs with scene overlap without the need to perform exhaustive putative matching and geometric verification. PAIGE achieves state-of-the-art performance and integrates well into existing Structure-from-Motion pipelines.

1. Introduction

Over the last years, large-scale Structure-from-Motion (SfM) has seen tremendous evolution in terms of robustness and speed in all stages of processing [1, 39, 13, 11, 42, 41, 37, 20]. Incremental SfM (Figure 2) commonly starts with feature detection and extraction (Stage 1), followed by matching (Stage 2), and geometric verification (Stage 3) of successfully matched pairs. After the matching and verification stage, typical SfM seeds the model with a carefully selected initial two-view reconstruction, before incrementally registering new cameras from 2D-3D correspondences, triangulating new 3D features, and refining the reconstruction using bundle-adjustment (Stage 4).

Generally, major computational effort is spent on Stages 2–4. In Stages 2 and 3, it is essential to discover a sufficient number of image correspondences that link together all parts of the scene to obtain complete and large-scale reconstructions. In addition, robust and accurate alignment is aided by finding multiple redundant image-to-image connections across the entire scene. However, exhaustively searching for these overlapping pairs is infeasible for large-scale image collections due to quadratic computational complexity in the number of images and features. Moreover, as the number of registered images grows, the scalability of bundle-adjustment algorithms becomes a significant performance bottleneck.

This paper evaluates existing techniques for reducing the cost of Stages 2 and 3, feature matching and geometric verification. Usually, the majority of image pairs in unordered Internet photo-collections do not have scene overlap, so rejecting those pairs dominates execution time, even though such pairs are not useful for 3D reconstruction. Consequently, various approaches have been proposed to efficiently find overlapping pairs in noisy datasets and only forward those pairs to Stages 2 and 3. A downside of sending fewer image pairs to Stages 2 and 3 is that enough images with overlapping geometry must be processed to produce accurate camera alignment and complete reconstructions. Hence, it is essential to find the right trade-off between computational efficiency and sufficient image connectivity.

Despite the impressive progress in reducing the cost of the matching (Stage 2), relatively little attention has been paid in comparing the techniques. The goals of this paper are therefore twofold: First, we present a comprehensive analysis and evaluation of various state-of-the-art matching techniques; second, we use the insights gained from this evaluation to propose the PAirwise Image Geometry Encoding (PAIGE) to build a scalable framework (Figure 1) for the efficient recognition of the relative viewing geometry, all without explicit feature matching and without reconstructing the actual camera configuration using geometric verification. The proposed encoding is based on location and orientation properties efficiently inferred from approximate feature correspondences. A subsequent classification strategy leverages the encoding to only perform matching and geometric verification for image pairs that are identified as overlapping. As demonstrated in comprehensive experiments, this novel approach leads to a further speedup of large-scale SfM than the existing state of the art.

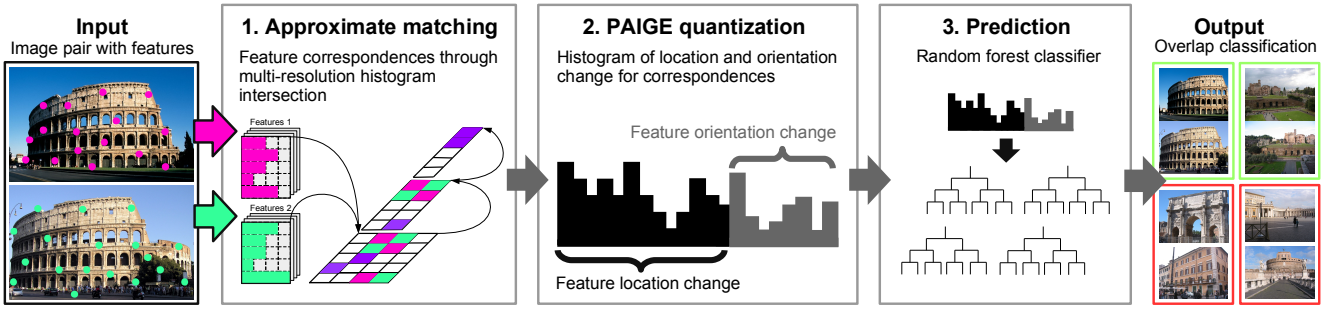


Figure 1. The proposed framework for PAIGE extraction, and its application for scene overlap and viewpoint change prediction.

2. Related work

Large-scale SfM systems have tremendously advanced in terms of increased robustness and reduced runtime. A variety of methods have been proposed to improve the efficiency in different stages of SfM pipelines (Figure 2).

Stage 1 While SIFT [27] is a popular choice for robust feature detection and description, the slightly more efficient SURF features are a commonly used alternative [5]. In addition, a number of binary features have also been proposed [36, 24, 3]. These binary features lead to a significant speedup of the extraction and the subsequent matching stage as well as a reduced memory footprint.

Stage 2 Various methods have been proposed to reduce the number of image pairs considered in the matching module. Frahm *et al*. [13] leverage iconic image selection through GIST clustering to find similar images. Agarwal *et al*. [1] employ image retrieval systems [30] to only match against similar images and then use approximate nearest neighbor feature matching. Furthermore, Chum *et al*. advance in the field of efficient image retrieval [10] and improve retrieval results with a randomized data mining method [9]. Another improvement to retrieval systems was developed by Chao *et al*. [7], who employ an online learning strategy to rerank retrieval results. Krapac et al. [23] and Jégou et al. [22] encode spatial information of features in bag-of-words models as used in retrieval systems. Orthogonally, Raguram *et al*. [35] use GPS tags to match images only to spatially nearby ones. Wu [42] follows a preemptive matching strategy by filtering image pairs that fail to match on a reduced feature set. Beyond that, Lou *et al*. [26] develop a scalable method to find connected components in large datasets. Most recently, Hartmann *et al*. [17] propose to predict the matchability of individual features to reduce the number of feature comparisons during feature matching; Havlena *et al*. [18] inspired by [32, 40] directly use the assignments of individual features to visual words in a vocabulary tree as verified correspondences for SfM, skipping the pairwise image matching stage altogether.

Stage 3 Apart from the advancements in fast essential matrix estimation [29], a number of efficient RANSAC [12] variants have been developed [28, 8, 33]. Complementary, Raguram *et al*. propose to improve the efficiency of geometric verification with an online learning approach [34].

Stage 4 Snavely *et al*. [39] compute skeletal subsets of images to reduce the runtime of incremental reconstruction, whereas Agarwal *et al*. [2] and Wu *et al*. [43] progress in the field of bundle-adjustment by developing efficient and scalable algorithms for multi-core machines. Complementary to the efforts in incremental SfM, Gherardi *et al*. [14] propose a hierarchical SfM pipeline with balanced branching and merging. Sinha *et al*. [38] compute two-view reconstructions from vanishing points followed by efficient 3D model merging, while Crandall *et al*. [11] describe a replacement for traditional incremental SfM by finding a coarse initial solution for bundle-adjustment using a discrete-continuous optimization approach based on GPS initializations. Recently, Wilson *et al*. [41] propose to estimate camera translations by solving simplified lower-dimensional problems with epipolar geometry averaging.

For the comparative evaluation of matching techniques (Stage 2) in this work, we choose one popular representative of each family of approaches. The above described approaches can be categorized into three different families of approaches. The first family, approximate matching techniques, describe images as a whole and avoid exhaustive pairwise image matching [1, 13, 9, 10, 26, 7]. The second family, exhaustive matching techniques, try to either preemptively filter image pairs [42] or reduce the cost of feature matching [17]. The third family consists of approaches that try to avoid pairwise matching and verification altogether [18]. We rely on publicly available implementations of the methods; implementations that have already been successfully applied in large-scale 3D reconstruction. In the following, we briefly describe the chosen approaches; an evaluation of their performance on several large-scale datasets (Table 1) is given in Section 3.

Image retrieval has been extensively employed in large-scale SfM [1, 13, 26]. Hence, we use it as a representative of the first family. Image retrieval is often efficiently implemented using vocabulary trees [30], an instance of bag-of-words (BoW) models, which try to describe images as

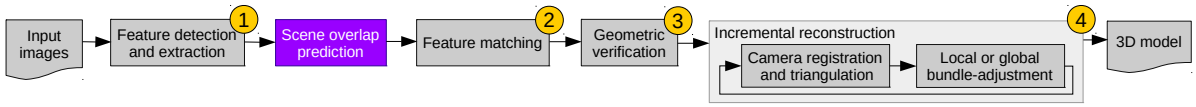


Figure 2. The proposed prediction framework (purple) integrated into a typical SfM pipeline.

a whole. Features are hierarchically quantized and indexed in the vocabulary tree. Similarity from indexed images to a query image is measured using, *e.g.*, tf-idf, co-occurrence, or burstiness scoring. In large-scale SfM systems, vocabulary trees are leveraged to match every image only against a number of most similar images (approximate nearest neighbors), effectively eliminating the quadratic computational cost in the number of images of exhaustive pairwise matching. The number of retrieved images is determined by retrieving a fixed number of images N_R and/or thresholding the similarity score. However, BoW similarities are noisy, due to faulty quantization and feature detection. As a consequence, it is difficult to find good similarity thresholds, which is why, in our analysis, we retrieve a fixed number of nearest neighbors per query image. We employ the implementation of Agarwal *et al*. [1], a tf-idf weighted vocabulary tree using min-distance metric and 1M visual words (branching factor 10, depth 5) trained from approximately 100M features (unrelated to evaluation datasets). We denote this method as *Retrieval* N_R.

Preemptive matching, as a representative of the exhaustive matching techniques, follows the idea that matching a small subset of the features is effective in determining whether an image pair has overlap. The method assumes that features detected at higher scales are more repeatable and stable across images; hence, if a small number N_P of L_P higher scale features match, the image pair is said to have overlap. Full putative feature matching is only performed for those pairs that pass this preemptive filtering stage. On the one hand, this strategy theoretically allows us to find all possible image pairs. On the other hand, it still has quadratic computational complexity in the number of features and images. However, the work for individual image pairs dramatically decreases (*e.g.* by a factor of 10,000 when $L_P = 100$), since feature matching is itself quadratic in the number of features. We use the implementation of Wu [42] and denote it as *Preemptive* N_P, setting $L_P = 100$, as suggested by the author.

Vocabulary matching, as a representative of the third family, skips the pairwise image and feature matching stages altogether by using the indexing of multiple features to the same visual word in a precomputed vocabulary tree as implicit matches. Feature matches between image pairs are then generated by the pairwise combination of all assigned features per visual word. A symmetric clustering matrix is used to find connected components in an image collection. To avoid ambiguous matches, only one visual word may appear in each image. For reasons of efficiency and to reflect the importance of a visual word w.r.t. frequency of its occurrence (similar to the motivation of tf-idf weighting), the method discards visual words that appear in too many images (the authors propose a threshold of 1%). Given a sufficiently large visual vocabulary, correspondences from assignments of features to visual words are stronger than from pairwise putative matching. Note that this method requires significantly more visual words than in standard vocabulary trees in order to achieve good performance. In addition, the proposed approach is infeasible for very large image collections with millions of images, since the clustering matrix cannot be stored in memory, as noted by the authors. We use the implementation and visual vocabulary provided by Havlena *et al*. [18], and denote the method as *VocMatch*.

	Images	Pairs	Verified pairs
London Eye	7,047	24,826,581	319,591 (1.29%)
San Marco	7,792	30,353,736	237,130 (0.78%)
Tate Modern	4,813	11,580,078	119,483 (1.03%)
Time Square	6,426	20,643,525	140,193 (0.68%)
Trafalgar	6,981	24,363,690	285,022 (1.17%)
Rome	16,179	130,871,931	– (–)

Table 1. Evaluation datasets.

3. Evaluation

In this section, we evaluate the previously described approaches on different large-scale datasets (Table 1). We propose to formulate the problem of finding overlapping image pairs as a classification problem, where we try to learn a model that separates image pairs with scene overlap (positive) from image pairs without scene overlap (negative). The objective of an optimal method is to minimize the ratio of false over true positives (overhead), whereas the true positives should comprise all relevant image pairs of the dataset. Since in Internet photo collections typically only a small fraction of the images are relevant and therefore an even smaller fraction of image pairs actually match, the effective runtime of a method is determined by the overhead. Hence, the goal of an optimal matching strategy is to produce minimal overhead while finding all true positives. In the end, the effective utility of a matching method for SfM is related to the completeness and stability of the resulting reconstructions.

The evaluation datasets comprise five crowd-sourced image collections (London Eye, San Marco, Tate Modern, Time Square, and Trafalgar) [7], and a well-studied dataset of Rome [25]. These collections contain a diverse set of

viewpoints, rather than a single dominant one. The first five datasets are contaminated with a large number of irrelevant images that do not match to the actual landmarks. Contrary, the Rome dataset only consists of relevant images, which should register to at least one landmark. For all experiments, we use SIFT features (Hessian-Affine [31] for *VocMatch*, Difference of Gaussian for all other methods). We consider an image pair as geometrically verified (*i.e.* it has scene overlap) if the putative SIFT matches (max. distance ratio of 0.8 between top two matches, max. cosine distance of 0.7, and mutual best matching) have at least 20 inliers in essential matrix estimation with RANSAC (4px Sampson error threshold). As a baseline approach, we exhaustively compute the ground-truth image pairs, with N_G denoting the number of verified pairs. The performance of each method is quantified in several measures obtained from the confusion matrix (N_{TP}: true positives, N_{FP}: false positives). First, we measure how many of the ground-truth image pairs are found (N_{TP}/N_G). Second, we measure the overhead of finding these pairs (N_{FP}/N_G). Third, we measure the required time by isolating the runtime of the respective method including the subsequent Stages 2 and 3. For the matching procedure, we use an optimized GPU implementation, and for geometric verification a multi-threaded RANSAC CPU implementation. To quantify the impact of the reduction of each method, we measure the completeness of 3D reconstruction in terms of the total number of registered cameras. All experiments were performed on the same machine with 2x12 physical cores, 256GB RAM, and a NVIDIA GeForce GTX TITAN Z graphics card. I/O overhead is excluded from the timing for all methods.

The results of the experiments are summarized in Table 2 and Figure 6. All methods significantly reduce the runtime and number of evaluated pairs compared to exhaustive matching. But they also produce a significant number of false negatives, *i.e.* they eliminate correct image pairs. Nevertheless, SfM is still able to produce quality reconstructions with the number of registered images being related to the number of verified image pairs. In this regard, we also observe that the number of registered images and, qualitatively, the stability of the reconstructed models saturates at some point. In other words, SfM does not substantially gain from finding all true positives. In the following, we briefly discuss the individual results of each approach.

Retrieval As can be seen from Table 2, vocabulary trees work relatively well in terms of precision, when only retrieving a few nearest neighbors. However, when more images are retrieved, such methods tend to yield many false positives, resulting in a large computational overhead in matching. Otherwise, in case only a few images are retrieved, the overhead of indexing and querying images in the vocabulary tree becomes more relevant to the overall runtime. While theoretically possible [1], it is comparatively challenging to efficiently scale the indexing and querying of a vocabulary tree across distributed machines for large-scale datasets.

Preemptive Due to quadratic feature matching cost, we must limit N_P to a low number for reasons of efficiency. Consequently, the threshold L_P must be chosen very low ($L_P = 4, N_P = 100$ as proposed by Wu [42]) to find relevant pairs. Thus, a small change in L_P has great effect on the performance of this filtering strategy – both in terms of efficiency and precision (compare *Preemptive 3* and *Preemptive 4*). Moreover, a small subset of the features may not adequately represent the entire image, resulting in a noisy classifier. Beyond that, image pairs with small overlap (*e.g.*, due to large scale change or different viewpoint centers) will likely fail to pass the filtering, because of the the low number of features, which may be spatially distributed across the entire image. This method can be relatively easily scaled across multiple cores and distributed machines.

VocMatch While this method drastically speeds up the computation of pairwise matching, it also makes some assumptions about the underlying structure of the image collection. Since the method discards highly frequent visual words, image collections of popular landmarks with many redundant viewpoints may produce less stable reconstructions due to the lack of long feature tracks. Moreover, the length of feature tracks depends on the relation of the number of features in the dataset and the codebook size of the vocabulary tree. We find that the track lengths of 3D points during reconstruction are significantly shorter for *VocMatch* than for the other methods, *i.e.* the estimated point locations are more uncertain. Setting the maximum frequency of visual words and using the right codebook size is difficult because there is usually no a priori knowledge about the distribution of images in crowd-sourced datasets. We can observe the impacts of the a priori assumptions by considering the high variance of the performance across the different datasets.

Summarizing the above evaluation, we conclude, that there is no need to find all true positive image pairs to produce good reconstructions in terms of stability and completeness. In fact, we only need a comparatively small fraction of the ground-truth image pairs. While the true positive rate accounts for some of the runtime, the methods' overall runtimes are mostly determined by the overhead, since RANSAC is especially expensive for false positive pairs. Moreover, scalability becomes especially important for large-scale datasets. None of the existing approaches provides sufficient recall with low overhead and fast runtime to produce quick reconstructions for large datasets. Considering this analysis, we propose an efficient and scalable learning-based approach to preemptively predict the geometric relation between an image pair with low overhead (*i.e.* low false positive rate). To represent the image

with a larger subset of features, we develop a hierarchical, approximate matching scheme that reduces the quadratic to amortized linear complexity. Based on the resulting implicit feature correspondences, we compute the PAirwise Image Geometry Encoding (PAIGE). A subsequent classification procedure leverages the encoding to predict whether an image pair has overlap or not. Finally, standard putative matching and geometric verification is only performed for image pairs that are predicted to overlap.

4. Pairwise image geometry

In this section, we develop PAIGE, a new approach for quickly predicting whether two images have scene overlap. Toward this goal, we begin by analyzing how pairwise image geometry relates to the pattern of correspondences between feature points in two images (Section 4.1). To use this intuition in a fast approach, we first hash all the features in an image into a fixed-sized data structure (Section 4.3), and then compute an approximate descriptor of the pattern between corresponding feature points (Section 4.4). The PAIGE approach works by learning to predict scene overlap from this representation. The whole process, from hashing the descriptor of approximate correspondences to evaluating the classifier, is linear in the number of features per image (Section 4.5), and is relatively light-weight in terms of actual computation. The hashing process preserves enough information about the geometry between pairs of images to allow PAIGE to produce accurate and fast predictions about whether two images should be sent onward to the computationally more expensive putative matching stage.

4.1. Feature correspondence and pairwise geometry

We define pairwise image geometry as the relative motion between two images. The relative motion between an image pair can be determined up to unknown scale by estimating the essential matrix [16] for a freely moving and by a homography for a purely rotating camera. Hence, an image pair has scene overlap, if we can estimate its relative motion from corresponding feature points.

Traditional SfM systems require the extraction of sparse image features (Stage 1), preferably invariant under radiometric and geometric transformations. Current practice uses local features that estimate four properties [19]: location $\{\bar{x}, \bar{y}\}$, orientation o, scale s, and the descriptor $\mathbf{f}$.

The central observation underlying the PAIGE approach is, that when images of the same structure are taken from different viewpoints, corresponding features change in scale, location, and rotation in recognizable patterns. Figure 3 visualizes patterns in the changes between features in a synthetic experiment, demonstrating the relation of pairwise geometry and the properties of corresponding features.

To produce Figure 3, we find feature correspondences for a pair of rendered images of a reference pattern with 256 feature points, with the first camera held stationary and the viewpoint of the second camera increasingly transformed. For this image pair, we calculate the displacement for each feature using normalized image coordinates, such that $\{x, y\} \in [0, 1]$ (to handle zoom), and measure rotation of features in degrees. Next, histograms quantize the distribution of these two measures of feature transformation. We observe, that the location change histogram (Figure 3 (a)) is sufficient to recognize purely translational camera motion. However, the location change alone does not distinguish between purely rotational motion (Figure 3 (b)) and a combination of translational and rotational camera motion (Figure 3 (c)). Considering histograms of both feature location and orientation change allows us to separate the two cases.

Based on this motivation, the following sub-sections describe how to efficiently find approximate feature correspondences and then to leverage estimates of the corresponding location and orientation changes to predict whether an image pair has scene overlap.

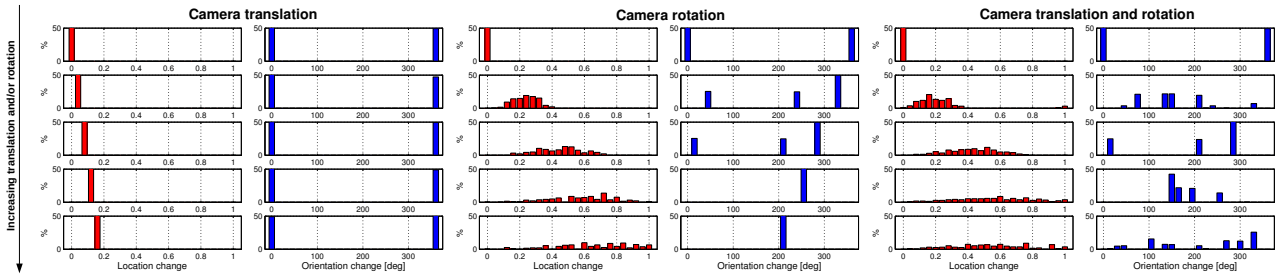
4.2. Approximate feature transformations

Computing the histograms shown in Figure 3 used knowledge of the exact feature correspondences between a pair of images. Our approach approximates this correspondence. Conceptually, we make two levels of relaxations to perform this approximation. First, instead of the exact correspondence, we can consider the motion (translation and rotation) between a feature in one image and any very similar features in the other image. As the next relaxation, we can consider individual dimensions of a feature descriptor. Each time two descriptors match in one dimension, we use their translation and rotation to increment the appropriate bins of the translation and rotation histograms. This later relaxation seems as if it would introduce a large number of spurious increments to the histograms, but because non-matching features rarely agree on many dimensions, unlike closely matching features, the noisy additions are spread out. In practice, this representation works well, as shown in the experiments (Section 7). Furthermore, this approach allows computing the histogram of translations and rotations from approximate correspondences to be done in two stages: 1) Hashing all of the features in an image into one fixed-size data structure. 2) Using two of these data structures to compute an approximate histogram of translations and rotations between features in two images.

4.3. Hashing the features from one image

Consider a collection (the images), $\mathbf{X} = \{\mathbf{F}_1, \ldots, \mathbf{F}_m\}$, of sets (the features for images) $\mathbf{F}_i = \{\mathbf{f}_1, \ldots, \mathbf{f}_{n_i}\}$ of cardinality n_i of d-dimensional feature descriptors $\mathbf{f}_j$. For simplicity, we assume that all entries, f_k, in each $\mathbf{f}_j$ are non-negative, and that $\|\mathbf{f}_j\|_2 = 1$. We quantize a given $\mathbf{F}$ into

$$\mathcal{F}(\mathbf{F}) = [\mathbf{H}_0(\mathbf{F}), \mathbf{H}_1(\mathbf{F}), ..., \mathbf{H}_{r-1}(\mathbf{F})] \quad (1)$$



(a) (b) (c)

Figure 3. SIFT feature location and orientation change histograms for (a) camera translation, (b) camera rotation $[0^\circ; 150^\circ]$ around the viewing direction, and (c) camera translation and rotation. Histograms from top to bottom with increasing translation and/or rotation.

as a concatenation of r differently weighted 2-dimensional multi-resolution histograms $\mathbf{H}_i(\mathbf{F}) \in \mathbf{M}_{d \times b_i}(\mathbb{R})$. Each $\mathbf{H}_i$ has 1-dimensional histograms with b_i bins for each of the d dimensions of the feature vectors $\mathbf{f}_j$, hence is $d \times b_i$-dimensional. The 1-dimensional histograms span the space $f_k \in [0, 1]$ using $b_i = 2^i$ bins of width $\Delta b = 2^{-i}$. To populate the histograms, each $\mathbf{f}_j \in \mathbf{F}_i$ contributes its assigned weight η (which varies depending on the task, see below) once to each of the d locations in each $\mathbf{H}_i$. Overall, the hashed descriptor, $\mathcal{F}(\mathbf{F}_i)$, for a set of features $\mathbf{F}_i$ from an image, has dimension $d \sum_{i=0}^{r-1} 2^i$ that does not depend on the number of feature descriptors, n_i, for the image.

This representation can be leveraged to establish approximate correspondences between two entities $\mathbf{F}_a$ and $\mathbf{F}_b$ by intersecting their respective $\mathcal{F}(\mathbf{F}_a)$ and $\mathcal{F}(\mathbf{F}_b)$. The more similar two features $\mathbf{f}_a \in \mathbf{F}_a$ and $\mathbf{f}_b \in \mathbf{F}_b$ are, the more they will contribute to corresponding bins in $\mathcal{F}(\mathbf{F}_a)$ and $\mathcal{F}(\mathbf{F}_b)$.

This approach is potentially prone to over-estimating the number of correspondences, since it finds matches separately in all marginals of $\mathbf{f}$, which might result in duplicate and false matches. However, if a pair of feature vectors $\mathbf{f}$ are very close (*e.g.* for a true correspondence), they will agree in more dimensions than dissimilar features. As the dimension of the descriptors increases this effect becomes stronger. Section 4.4 explains, how we account for the differing similarity across levels by weighting the relevance of correspondences based on the histogram resolution. The approximate matching scheme can naturally deal with sets of unequal cardinalities, since a feature in the smaller set is implicitly matched to multiple features in the larger set.

This scheme borrows ideas from the pyramid match approach [15], but differs in a fundamental way. The pyramid match approach treats the descriptor vector as a whole, and in practice it is therefore often implemented using a sparse histogram. In our approach, we hash each dimension of the descriptor separately, resulting in a more efficient, fixed-size histogram implementation. Moreover, the traditional pyramid match approach intersects the raw counts of overlapping features, while we use weighted histograms.

In the next section, we will see how to use this weighted matching scheme to encode the geometric properties in the PAIGE descriptor.

4.4. PAIGE quantization

The PAirwise Image Geometry Encoding (PAIGE) is defined as the function

$$\mathcal{P} \colon \mathbf{X} \times \mathbf{X} \to \mathbb{R}^{d_\mathcal{P}} \tag{2}$$

and quantifies the distribution of location and orientation changes between an image pair $(\mathbf{F}_a, \mathbf{F}_b)$ based on its feature correspondences $(\mathbf{f}_a \in \mathbf{F}_a, \mathbf{f}_b \in \mathbf{F}_b)$. The approximate matching scheme described in Section 4.3 is used to implicitly establish these correspondences. Therefore, we compute separate multi-level histograms $\{\mathcal{F}_x, \mathcal{F}_y, \mathcal{F}_o, \mathcal{F}_1\}$ in a computationally efficient manner for the respective cases $\eta \in \{x, y, o, 1\}$. In other words, we quantize the geometric information of a single image in separate histograms, and count the number of elements (the features) per bin with $\eta = 1$. The image locations are normalized using the dimensions of the image, such that

$$\Delta x \in [-1, 1]\,, \Delta y \in [-1, 1]\,, \Delta o \in [-2\pi, 2\pi]\,. \tag{3}$$

In the next step, we average the location and orientation histograms to account for the fact that multiple features might populate the same bin

$$\overline{\mathcal{F}} = \left[\frac{\mathcal{F}_x}{\mathcal{F}_1}, \frac{\mathcal{F}_y}{\mathcal{F}_1}, \frac{\mathcal{F}_o}{\mathcal{F}_1} \right] \tag{4}$$

For all pairwise combinations of images $(\mathbf{F}_a, \mathbf{F}_b)$ in $\mathbf{X}$, we can thereby efficiently calculate the approximate change in location and orientation per marginal bin as

$$\Delta\overline{\mathcal{F}} = \overline{\mathcal{F}}_a - \overline{\mathcal{F}}_b \tag{5}$$

We describe the distribution of these location and orientation changes using the PAIGE feature, which is defined as a concatenation of uniformly spaced, weighted histograms

$$\mathbf{h}\left(\mathbf{F}_a, \mathbf{F}_b\right) = [\mathbf{h}_{\Delta x}, \mathbf{h}_{\Delta y}, \mathbf{h}_{\Delta o}] \tag{6}$$

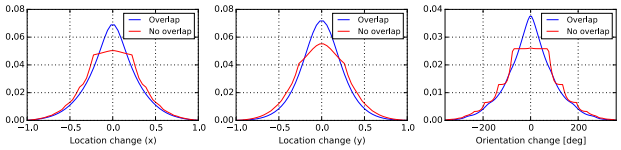


Figure 4. Average PAIGE from London Eye dataset, separated into location and orientation parts.

populated from $\Delta\overline{\mathcal{F}}$. The dimensionality of PAIGE is

$$d_{\mathcal{P}} = d_{\Delta x} + d_{\Delta y} + d_{\Delta o} \tag{7}$$

since $\mathbf{h}_{\Delta x} \in \mathbb{R}^{d_{\Delta x}}$, $\mathbf{h}_{\Delta y} \in \mathbb{R}^{d_{\Delta y}}$, $\mathbf{h}_{\Delta o} \in \mathbb{R}^{d_{\Delta o}}$. Finally, the encoding is normalized to achieve invariance w.r.t. the number of feature correspondences

$$\mathcal{P}\left(\mathbf{F}_a, \mathbf{F}_b\right) = \frac{\mathbf{h}\left(\mathbf{F}_a, \mathbf{F}_b\right)}{\left\|\mathbf{h}\left(\mathbf{F}_a, \mathbf{F}_b\right)\right\|_2} \tag{8}$$

The weight ω a populated bin in $\mathcal{F}$ contributes to PAIGE depends on the similarity of the approximate correspondences. As shown in Section 4.3 the similarity of a match is dependent on the resolution and thus the level i of the histogram $\mathbf{H}_i$ in $\mathcal{F}$. Hence, we choose the weight as $\omega_i = 2^{i-r}$. PAIGE is naturally robust against mismatches, since it is dominated by fine-grained correspondences in the higher-resolution histogram levels. Additionally, it is able to capture the overall location and orientation changes through the correspondences in the coarser histogram levels. PAIGE is intentionally designed as a non-symmetric function, *i.e.* $\mathcal{P}\left(\mathbf{F}_a, \mathbf{F}_b\right) \neq \mathcal{P}\left(\mathbf{F}_b, \mathbf{F}_a\right)$, since this allows us to encode the direction of relative camera motion.

4.5. Efficiency

The described matching approach enables us to find approximate feature correspondences without performing exhaustive pairwise feature matching, which is quadratic in the number of features $O(n^2)$. More precisely, the population of $\mathcal{F}$ is $O(drn)$, since the d marginals of $\mathbf{f}$ contribute to a maximum of r histograms. The normalization step and the PAIGE quantization are performed for each element in $\mathcal{F}$, and thus are $O(2^{r+1}d)$. Typically, it is $n \gg r$ and $n \gg d$. Hence, the amortized computational complexity of quantizing PAIGE is $O(n)$. Note that we hash every image in a collection in the fixed-sized data structure $\mathcal{F}$ independently, and then reuse it for the exhaustive pairwise computation of PAIGE to reduce the computational effort.

5. Classification

Based on the proposed PAIGE feature (Equation 8), we next design a binary classifier to predict scene overlap. In doing so, we try to learn a model that separates image pairs with scene overlap (positive) from image pairs without scene overlap (negative). Choosing a suitable classifier depends on two main factors. First, the joint distribution of location and orientation change is expected to be complex over the complete space of possible pairwise image configurations. Hence, we need a classifier that is able to discriminate this complex parameter space. Second, the main motivation for the proposed method is a speed improvement over the traditional approach of exhaustive feature matching and geometric verification; therefore, the classifier should require minimal computational effort for maximal benefit. In our experiments, random forests [21, 6, 4] gave the best results in terms of accuracy and computational efficiency.

We use SIFT to extract invariant features at different scales. Note, any other invariant features could be employed alternatively. The 128-dimensional descriptors $\mathbf{f}$ are normalized and stored with 8-bit precision. We use $r = 9$ as the number of multi-resolution histograms; the number of bins of the finest-resolution histogram therefore equals the descriptor discretization. Empirically, the dimensionality of PAIGE is chosen as $d_{\Delta x} = d_{\Delta y} = 50$ and $d_{\Delta o} = 100$.

6. Training

Large-scale Internet photo-collections from several different landmarks across the world and a set of sequential image sequences acquired by mobile video cameras (to counter the orientation bias of crowd-sourced images) serve as the dataset for training the random forest classifier. Note, the training dataset is disjoint from the evaluation datasets. Ground-truth data is extracted by exhaustive pairwise image matching and subsequent geometric verification for approximately 30M image pairs. Then, PAIGE is extracted for all image pairs a and b in the forward $\mathcal{P}\left(\mathbf{F}_a, \mathbf{F}_b\right)$ and backward $\mathcal{P}\left(\mathbf{F}_b, \mathbf{F}_a\right)$ directions. Hence, for each image pair, we generate two training samples with the same label. Analogously, when we classify an image pair, we can extract the forward and backward PAIGE features with requiring only small additional computational overhead, since we need only invert the order of subtraction in Equation 5. We then classify both features and use the more confident prediction as the final classification result. Due to the fact that most image pairs in unordered collections do not have scene overlap, we reduce (via random sub-sampling) the number of negative samples with the goal of training classifiers with differently tuned properties in terms of the expected overhead. We denote these versions as *PAIGE* $N_{\mathcal{P}}$, where $N_{\mathcal{P}}$ is the ratio of negative over true training samples. Using 3-fold cross-validation, we determined design choices for the classifier, including using a forest with 50 decision trees, entropy as the splitting criterion, and considering all features when searching for the best split at each node in a tree. A minimum number of three samples per leaf is enforced to avoid over-fitting.

	Time	Prec.	Found	Overhead	Reg. images
Retrieval 25	**17h24m**	**0.22**	**0.13**	**3.59**	**11845**
London Eye	3h58m	0.29	0.14	2.40	2354
San Marco	3h41m	0.28	0.17	2.55	3392
Tate Modern	2h30m	0.18	0.12	4.63	1429
Time Square	3h16m	0.14	0.12	6.01	2014
Trafalgar	3h59m	0.17	0.11	4.85	2656
Rome	6h43m	–	–	–	15412
Retrieval 50	**28h39m**	**0.18**	**0.21**	**4.68**	**13544**
London Eye	5h25m	0.25	0.24	2.99	3280
San Marco	5h48m	0.23	0.28	3.41	3703
Tate Modern	3h28m	0.14	0.20	6.01	1481
Time Square	6h57m	0.11	0.18	8.34	2169
Trafalgar	7h0m	0.14	0.18	6.40	2911
Rome	9h4m	–	–	–	15366
Retrieval 100	**61h34m**	**0.14**	**0.32**	**6.39**	**14531**
London Eye	11h55m	0.20	0.37	3.95	3331
San Marco	17h7m	0.17	0.41	4.87	3929
Tate Modern	6h32m	0.11	0.31	8.01	1647
Time Square	16h6m	0.08	0.26	11.86	2463
Trafalgar	9h54m	0.10	0.26	8.75	3161
Rome	17h41m	–	–	–	15388
Preemptive 3	**164h41m**	**0.05**	**0.32**	**17.59**	**14767**
London Eye	38h6m	0.08	0.36	11.14	3418
San Marco	42h6m	0.06	0.34	16.24	3815
Tate Modern	14h33m	0.07	0.35	13.56	1825
Time Square	34h46m	0.03	0.24	34.53	2401
Trafalgar	35h10m	0.04	0.38	22.00	3308
Rome	223h12m	–	–	–	15401
Preemptive 4	**58h26m**	**0.13**	**0.21**	**6.48**	**14694**
London Eye	13h32m	0.21	0.25	3.83	3477
San Marco	13h58m	0.16	0.23	5.23	3744
Tate Modern	6h28m	0.18	0.23	4.61	1930
Time Square	10h35m	0.06	0.13	14.84	2351
Trafalgar	13h53m	0.09	0.25	9.57	3192
Rome	80h43m	–	–	–	15298
VocMatch	**11h15m**	**0.32**	**0.24**	**2.17**	**4247**
London Eye	2h15m	0.30	0.43	2.35	1353
San Marco	2h52m	0.29	0.37	2.47	1474
Tate Modern	1h21m	0.39	0.17	1.59	637
Time Square	2h46m	0.28	0.05	2.60	316
Trafalgar	2h1m	0.76	0.10	0.32	467
Rome	7h48m	–	–	–	12944
PAIGE 10	**84h31m**	**0.11**	**0.36**	**8.47**	**13520**
London Eye	18h0m	0.13	0.35	6.70	3167
San Marco	20h25m	0.10	0.36	9.25	3544
Tate Modern	9h54m	0.14	0.45	6.17	1490
Time Square	4h31m	0.23	0.28	3.38	2193
Trafalgar	31h42m	0.07	0.48	13.07	3126
Rome	83h55m	–	–	–	15298
PAIGE 20	**24h15m**	**0.29**	**0.27**	**2.44**	**12198**
London Eye	6h28m	0.30	0.28	2.33	2905
San Marco	4h22m	0.35	0.26	1.83	3145
Tate Modern	3h42m	0.30	0.35	2.37	1338
Time Square	1h8m	0.92	0.23	0.09	1999
Trafalgar	8h36m	0.18	0.32	4.41	2811
Rome	22h49m	–	–	–	14725
PAIGE 30	**7h18m**	**0.63**	**0.15**	**0.59**	**10697**
London Eye	1h35m	0.81	0.16	0.24	2508
San Marco	1h20m	0.84	0.15	0.19	2770
Tate Modern	0h55m	0.70	0.18	0.42	1204
Time Square	0h43m	0.99	0.14	0.01	1747
Trafalgar	2h44m	0.35	0.18	1.84	2468
Rome	5h14m	–	–	–	14566

Table 2. Precision ($N_{TP}/(N_{TP}+N_{FP})$), found pairs (N_{TP}/N_G), overhead (N_{FP}/N_G), and number of registered images.

7. Evaluation of PAIGE

We perform the same experiments for PAIGE as for the other methods. The PAIGE feature for overlapping vs. non-overlapping pairs is shown in Figure 4. The results in Table 2 show, that *PAIGE 30* has the lowest false positive rate of any method and hence the lowest overhead in reconstruction cost, while still reconstructing nearly as much as any other technique. At the other end of the spectrum, *PAIGE 10* has the highest true positive rate and results in nearly the highest reconstruction completeness at modest computational cost. Interestingly, PAIGE outperforms the other methods on the Time Square dataset, for which we find that the predictions of positives and negatives are much more separated and confident than for the other datasets. The clear separation is caused by the many video screens (dynamic scenes) and the day/night images, resulting in clearly incorrect pairwise geometry and sets of SIFT features that clearly cannot be aligned. The resulting models of PAIGE are stable and cover the entire scenes (Figure 5).

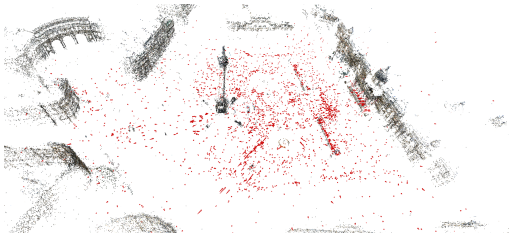


Figure 5. Reconstruction based on PAIGE for Trafalgar dataset.

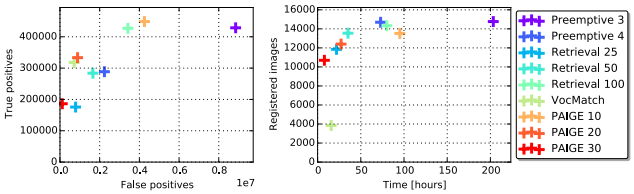


Figure 6. Visualization of the overall evaluation results.

8. Conclusion and outlook

In this paper, we conduct a comprehensive evaluation of state-of-the-art matching methods. Based on the insights of this evaluation, we propose PAIGE, a novel learning-based approach to identify overlapping image pairs for improved efficiency in the matching stage of SfM. We show, that approximate correspondence information reveals enough information to reliably predict the pairwise image geometry, resulting in significant speedups compared to traditional, exact correspondence approaches. Moreover, we show that learning-based methods can effectively support 3D reconstruction, in this case for improved efficiency. In future, it will be interesting to explore how to leverage PAIGE for other modules in SfM, *e.g.* to improve the robustness of incremental reconstruction by inferring geometric information between image pairs.

Acknowledgment This material is based upon work supported by the National Science Foundation under Grant No. IIS-1252921, IIS-1349074, IIS-1452851, CNS-1405847, and by the US Army Research, Development and Engineering Command Grant No. W911NF-14-1-0438.

References

[1] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski. Building rome in a day. *Commun. ACM*, 54(10):105–112, 2011. 1, 2, 3, 4

[2] S. Agarwal, N. Snavely, S. Seitz, and R. Szeliski. Bundle adjustment in the large. In *Proc. ECCV*, volume 6312, pages 29–42. 2010. 2

[3] A. Alahi, R. Ortiz, and P. Vandergheynst. Freak: Fast retina keypoint. In *Proc. CVPR*, pages 510–517, 2012. 2

[4] Y. Amit and D. G. Y. Shape quantization and recognition with randomized trees. *Neural Computation*, 9:1545–1588, 1997. 7

[5] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In A. Leonardis, H. Bischof, and A. Pinz, editors, *Proc. ECCV*, volume 3951, pages 404–417. 2006. 2

[6] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. 7

[7] S. Cao and N. Snavely. Learning to match images in large-scale collections. In *Proc. ECCV*, volume 7583, pages 259–270. 2012. 2, 3

[8] O. Chum and J. Matas. Matching with prosac-progressive sample consensus. In *Proc. CVPR*, volume 1, pages 220–226, 2005. 2

[9] O. Chum and J. Matas. Large-scale discovery of spatially related images. *IEEE PAMI*, 32(2):371–377, 2010. 2

[10] O. Chum, A. Mikulik, M. Perdoch, and J. Matas. Total recall ii: Query expansion revisited. In *Proc. CVPR*, pages 889–896, 2011. 2

[11] D. Crandall, A. Owens, N. Snavely, and D. P. Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In *Proc. CVPR*, 2011. 1, 2

[12] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981. 2

[13] J.-M. Frahm, P. Fite-Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y.-H. Jen, E. Dunn, B. Clipp, S. Lazebnik, and M. Pollefeys. Building rome on a cloudless day. In *Proc. ECCV*, volume 6314, pages 368–381. 2010. 1, 2

[14] R. Gherardi, M. Farenzena, and A. Fusiello. Improving the efficiency of hierarchical structure-and-motion. In *Proc. CVPR*, pages 1594–1600, 2010. 2

[15] K. Grauman and T. Darrell. The pyramid match kernel: discriminative classification with sets of image features. In *Proc. ICCV*, volume 2, pages 1458–1465, 2005. 6

[16] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Second edition, 2004. 5

[17] W. Hartmann, M. Havlena, and K. Schindler. Predicting matchability. In *Proc. CVPR*, 2014. 2

[18] M. Havlena and K. Schindler. Vocmatch: Efficient multiview correspondence for structure from motion. In *Proc. ECCV*, volume 8691, pages 46–60. 2014. 2, 3

[19] J. Heinly, E. Dunn, and J.-M. Frahm. Comparative evaluation of binary features. In *Proc. ECCV*, pages 759–773. 2012. 5

[20] J. Heinly, J. L. Schönberger, E. Dunn, and J.-M. Frahm. Reconstructing the World* in Six Days *(As Captured by the Yahoo 100 Million Image Dataset). In *Proc. CVPR*, 2015. 1

[21] T. K. Ho. The random subspace method for constructing decision forests. *IEEE PAMI*, 20(8):832–844, 1998. 7

[22] H. Jégou, M. Douze, and C. Schmid. Improving bag-of-features for large scale image search. *IJCV*, 87(3):316–336, 2010. 2

[23] J. Krapac, J. Verbeek, and F. Jurie. Modeling spatial layout with fisher vectors for image categorization. In *Proc. ICCV*, pages 1487–1494, 2011. 2

[24] S. Leutenegger, M. Chli, and R. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *Proc. ICCV*, pages 2548–2555, 2011. 2

[25] Y. Li, N. Snavely, and D. P. Huttenlocher. Location recognition using prioritized feature matching. In *Proc. ECCV*, pages 791–804, 2010. 3

[26] Y. Lou, N. Snavely, and J. Gehrke. Matchminer: Efficient spanning structure mining in large image collections. In *Proc. ECCV*, 2012. 2

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

[28] J. Matas and O. Chum. Randomized ransac with sequential probability ratio test. In *Proc. ICCV*, volume 2, pages 1727–1732, 2005. 2

[29] D. Nister. An efficient solution to the five-point relative pose problem. In *Proc. CVPR*, volume 2, pages II–195–202 vol.2, 2003. 2

[30] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Proc. CVPR*, pages 2161–2168, 2006. 2

[31] M. Perdoch, O. Chum, and J. Matas. Efficient representation of local geometry for large scale object retrieval. In *Proc. CVPR*, pages 9–16, 2009. 4

[32] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proc. CVPR*, 2007. 2

[33] R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J. Frahm. Usac: A universal framework for random sample consensus. *IEEE PAMI*, 35(8):2022–2038, 2013. 2

[34] R. Raguram, J. Tighe, and J.-M. Frahm. Improved geometric verification for large scale landmark image collections. In *Proc. BMVC*, pages 77.1–77.11, 2012. 2

[35] R. Raguram, C. Wu, J.-M. Frahm, and S. Lazebnik. Modeling and recognition of landmark image collections using iconic scene graphs. *IJCV*, 95(3):213–239, 2011. 2

[36] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *Proc. ICCV*, pages 2564–2571, 2011. 2

[37] J. L. Schönberger, F. Radenović, O. Chum, and J.-M. Frahm. From Single Image Query to Detailed 3D Reconstruction. In *Proc. CVPR*, 2015. 1

[38] S. N. Sinha, D. Steedly, and R. Szeliski. A multi-stage linear approach to structure from motion. In K. N. Kutulakos, editor, *Trends and Topics in Computer Vision*, volume 6554, pages 267–281. 2012. 2

[39] N. Snavely, S. M. Seitz, and R. Szeliski. Skeletal graphs for efficient structure from motion. In *Proc. CVPR*, 2009. 1, 2

[40] H. Stewnius, S. H. Gunderson, and J. Pilet. Size matters: Exhaustive geometric verification for image retrieval accepted for eccv 2012. In *Proc. ECCV*, pages 674–687. 2012. 2

[41] K. Wilson and N. Snavely. Robust global translations with 1dsfm. In *Proc. ECCV*, volume 8691, pages 61–75. 2014. 1, 2

[42] C. Wu. Towards linear-time incremental structure from motion. In *Proc. 3D Vision*, pages 127–134, June 2013. 1, 2, 3, 4

[43] C. Wu, S. Agarwal, B. Curless, and S. Seitz. Multicore bundle adjustment. In *Proc. CVPR*, pages 3057–3064, 2011. 2