Recent Developments in Large-scale Tie-point Search

Konrad Schindler, Wilfried Hartmann, Michal Havlena, Zürich

ABSTRACT

Matching corresponding point features in different images, which then serve as tie-points for camera orientation, is a basic step of the photogrammetric workflow. Increasingly one is faced with large, unordered image sets, e.g. from untrained users or even crowd-sourced from Internet photo collections. In such a setting, tie-point matching becomes a bottleneck of the orientation pipeline. On the one hand recording without detailed viewpoint planning implies a denser set of viewpoints with larger overlaps – and thus more image pairs – to ensure appropriate coverage and to a reliable reconstruction in spite of the ad-hoc network geometry. On the other hand, without a planned recording sequence it is not even known which images overlap. One thus faces the additional challenge to determine which pairs of images see the same part of the scene and should be fed into the matching step. The paper reviews recent developments in this field, which make it possible to generate tie-points and reconstruct unordered image sets with thousands of images.

1. INTRODUCTION

The switch to digital photography, and the continuing growth of computing power, storage, and transmission bandwidth have lead to important changes in the field of image-based 3D modeling. The amount of available image data has increased, and photogrammetric recording has spread to new application fields. At the same time improved processing methods nowadays allow one to generate fairly accurate reconstructions automatically, even from consumer cameras. Examples of this development include the photogrammetric use of small drones, as well as reconstruction from crowd-sourced images (Agarwal et al., 2009).

Photogrammetric object reconstruction consists of two steps, first image orientation in a common 3D coordinate frame (a.k.a. camera pose estimation), and second the generation of a dense point cloud or surface model. Her we are concerned with the first step. Surface orientation has also seen an impressive development (Hirschmüller, 2008; Hiep et al., 2009; Jancosek and Pajdla, 2011), but since it is always carried out locally for appropriate subsets of the oriented camera network, it is less affected by the dataset size. Figure 1 illustrates the 3D modeling pipeline. After acquiring images that cover the scene of interest, repeatable *interest points* are extracted from the input images and encoded with *descriptors* of the surrounding image patches. By comparing descriptors from different images, one finds point *matches* between image pairs. The matches serve as input for *pose estimation*, normally with a combination of pairwise relative orientation, tie-point triangulation, and spatial resection. Finally, the camera orientations and the sparse tie-point cloud are refined with bundle adjustment.

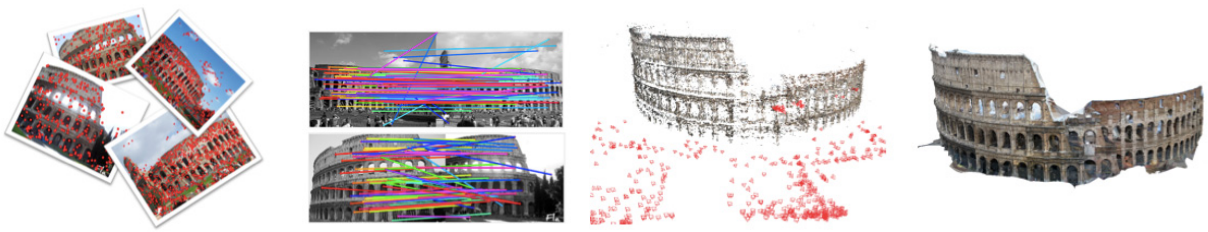

Figure 1: The 3D modeling pipeline. Interest points are extracted and matched to obtain tie-points. The tie-points enable (relative) image orientation. Finally a (optional) dense reconstruction recovers a point cloud or surface model.

In some of the new applications of photogrammetric modeling, one is confronted with large *unordered* image sets, meaning that it is not known in advance which images share a common field of view and can be matched. Examples include crowd-sourced imagery from the Internet; recordings from small micro aerial vehicles (MAVs) with low-quality navigation systems; but also close-range projects in general: untrained users have great difficulties to acquire images in a predefined pattern, and even experts are slowed down a lot by having to adhere to a strict recording protocol in complicated environments (e.g. industrial installations).

Traditionally, photogrammetry has preferred ordered image sets, where the image overlaps (or even the approximate orientations) are known before the processing starts; or unordered image sets with a small number of marked and easily detectable tie-points (e.g. high-contrast stickers) which can be disambiguated by their relative positions. Where available, such a procedure is often still preferable. Still, new applications make it necessary to deal also with unordered image sets, and in fact the ability to orient arbitrary images can also be useful in the traditional setting, e.g. to reduce field time and react quickly to unexpected difficulties on site. At the extreme end of the spectrum we find recent research in computer vision that aims to reconstruct 3D scenes from totally uncontrolled, crowd-sourced Internet images.

Modern projects can consist of thousands of unordered images (or sometimes millions, Heinly et al., 2015). Brute-force matching thus becomes intractable. To deal with situations where one can no longer compare every pair of interest point descriptors for every pair of images, there are three main possibilities: *(i)* reduce the matching time per image pair, by efficient data structures and by reducing the number of feature points per image; *(ii)* reduce the number of images, by finding those which are most important for reconstruction; and *(iii)* reduce the number of image pairs, either by finding those for which it is likely that tie points can be found or by sidestepping pairwise matching altogether.

2. IMAGE MATCHING

Tie-point generation starts with detecting interest points in each image separately. Usually these are corner features of the Harris/Förstner type or blob features like Difference-of-Gaussians (DoG). The computational cost of this step is normally negligible, since it must be done only once per image, and can be easily parallelized. Then a descriptor is computed for each interest point. The (pairwise) matching task consists in finding, for each descriptor in the source image, the most similar descriptor from the target image. (Dis)similarity is measured by a distance in the feature space, thus finding the most similar descriptor amounts to nearest-neighbor search. Beyond being the nearest neighbor, a valid match must usually fulfill additional criteria. Often there is a threshold for the maximum distance, but this turns out not to be overly effective, because of the high dimensionality of the descriptor space. A much stronger criterion, introduced by Lowe (2004), is to retrieve not only the nearest but also the second-nearest neighbor, and to compare the corresponding distances. If the second-best descriptor is almost as close as the best one, the matching is ambiguous and one might pick the wrong point, so the match is discarded. In practice exact nearest-neighbor search is inefficient in high dimensions, therefore the standard procedure is approximate nearest neighbor (ANN) search with *kD*-trees, or ensembles of such trees known as *kD*-forests.

2.1. Reducing the Number of Features

Finding approximate nearest neighbors instead of exact ones already makes pairwise matching more efficient. A second obvious way to speed up matching is to start with fewer interest points. Ideally, one would discard only points that later will not produce a successful match. Note, a good point

filtering heuristic will not only speeds up matching, but also will reduce the fraction of wrong matches, and this will benefit the subsequent orientation procedure, since robust pose estimation techniques like RANSAC are faster and less prone to failure with a cleaner tie-point set.

There are several possibilities to weed out key-points before the matching stage and still obtain useful tie-points. The first obvious idea is to raise the threshold of the interest point detector. By design a stricter threshold will return fewer points, but in general those which have the highest contrast, and thus the best repeatability (and localization accuracy). Importantly, the high detection score does not, per se, guarantee points with better matching potential. For example, vegetation under strong sunlight tends to generate many strong interest points, which are however very unlikely to become successful tie-point matches; whereas points with only moderate contrast like for example stone ornaments under diffuse lighting could be valuable tie-points, but will not survive a high detection threshold.

A second strategy at the level of interest point detection is to use only interest points from higher levels of the scale pyramid (stronger blur / larger feature scale) for tie-point matching (Wu, 2013). Empirically, most correspondences are found between points at nearby scales due to limited scale invariance, which means that one can hope to nevertheless obtain enough matches. The price to pay is on the one hand a higher uncertainty of the tie-point measurements, which are based on lower-frequencies of the image content; on the other hand there sometimes are only few low-resolution features, which additionally are often not well distributed. As the large majority of points are found at fine scales, one quickly is forced to use almost all pyramid levels, in which case the feature selection is close to random and loses a large portion of the matchable interest points.

A third, arguably more principled strategy is not to interfere with interest point detection. Instead, one lets the interest point detector find a large number of features and then examines the descriptors to predict which ones are likely to later generate a successful match. Since keeping points that later produce a match is the actual objective, one can expect such a prediction to outperform selection by the detector. As mentioned above, to be accepted as a tie-point match two descriptors should not only be nearest neighbors, but also must to have a distance below some threshold, and pass the second-best ratio test. The more important and much stricter filter is the second condition: if the nearest neighbor is significantly further away than the next best one, then the match is unambiguous. The important point here is that ultimately these tests determine which points become tie-points. It has therefore been proposed to learn a binary classifier, which can predict, for a single descriptor, how likely it is to pass the tests and generate a match (Hartmann et al., 2014). Like interest point detection the prediction can be done independently in each image, before nearest neighbor search. Compared to pruning at detector level the method does require an extra effort to compute the larger number of descriptors and evaluate the classifier, but that effort is negligible compared to the savings through fewer nearest-neighbor queries.

3. EFFICIENT MATCHING SCHEMES

The methods discussed so far reduce the time required to find the matches between two given images. They thus also apply for conventional, ordered image sets. In unordered image sets a second issue is even more important. Without knowledge about the camera layout one would in principle have to try all $N*(N-1)/2$ possible pairs of images, only to find out that the large majority of all pairs do not overlap and cannot be matched. Such a naïve approach is infeasible, and different strategies have been developed to avoid it.

3.1. Reduced Number of Image Pairs by Image Retrieval

To avoid wasted matching effort one needs to find the *connectivity* of the image set, i.e. which images share a part of their view-fields and can be matched. The connectivity can be formalized by a match graph, which has a node for each image and an edge between every pair of images that share a sufficient number of tie-points (Snavely et al., 2008). Constructing the exact match graph would again require exhaustive tie-point matching, instead one settles for an approximate match graph that can be found with a lot lower computational cost.

A first idea is to iteratively build an approximate match graph (Agarwal et al., 2009). Each iteration proposes a number of candidate edges, and then verifies them by matching the two associated images. Candidate edges connect visually similar images, and can be found with methods borrowed from image retrieval. To that end, the interest point descriptors are quantized to a fixed, discrete vocabulary of "visual words" (Sivic and Zisserman, 2003; Nister and Stewenius, 2006). Then, images can be represented as weighted histograms of visual words, so-called *tf-idf* (term frequency – inverse document frequency) vectors. The scalar product of two *tf-idf* vectors is an efficiently computable measure of image similarity. For a given image in the match graph, one can thus quickly find the M (typically about 10) most similar images and add the corresponding edges to the graph. The incremental construction of the match graph based on *tf-idf* vectors can also be interleaved with relative camera pose estimation and tie-point triangulation to gradually build up the camera network (Havlena et al., 2009).

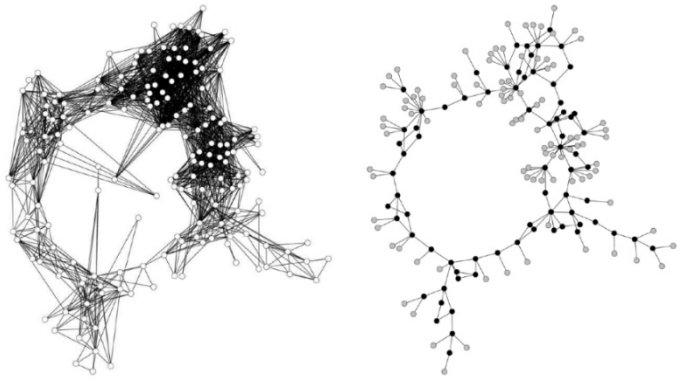

Figure 2: Reducing the full match graph (left) to a skeletal set (right) greatly reduces the number of image pairs that need to be matched, while nevertheless preserving an acceptable camera network for reconstruction. Figure courtesy of N. Snavely.

If image similarity is a good proxy for matching success, then the selection will generate large savings. However, similarity, respectively matching success, as such is not necessarily an indication that an image pair is useful for 3D reconstruction. Retrieval methods tend to find images with very similar viewpoints. These can of course be matched, but have insufficient baseline to support 3D modeling. This leads to a second group of methods, where one directly decimates the image set from which the pairs are selected.

3.2. Reduced Input Image Set

Unordered image sets, especially those collected from multiple photographers, often exhibit a very heterogeneous distribution. Some regions are highly redundant with a number of almost identical views, whereas in other regions the viewpoints are sparser and some images are very important to preserve the connectivity. If one manages to eliminate the redundant images while keeping the unique views, one can expect to obtain a much smaller image set, and thus efficient matching and

reconstruction. Note the slightly different objective: while retrieval-type methods aim to discard unmatchable image pairs, here one deliberately also discards many matchable pairs, in order to obtain the strongest possible reduction that still permits 3D reconstruction.

Figure 3: Unordered image sets can be highly redundant in some parts, therefore they can be automatically decimated by clustering. The figure shows example images from one cluster that can be represented with a single iconic view.

An obvious approach in this context is to cluster the input data by image similarity and keep only one image per cluster, the so-called "iconic view" (Li et al., 2008; Frahm et al., 2010). Similarity can again be measured with *tf-idf* distances, or, even more efficiently, with global image descriptors like GIST (Oliva and Torralba, 2001). Reconstruction then proceeds only with the iconics, optionally the remaining images can be added later. Empirically, larger clusters are often also cleaner, which can be exploited in subsequent steps.

A different way to cut down the number of pairwise matching operations is to construct an approximate match graph completely, e.g. by thresholding exhaustive pairwise *tf-idf* similarities, and then select a subset of nodes such that all images which are *not* members of that subset have a connection to at least one node who is a member (Havlena et al., 2010). If successful, this procedure will yield a small subset that has the necessary connectivity for pose estimation and preliminary reconstruction, such that one can register the remaining images to the network later. The search for the subset is an instance of a well-studied graph-theoretic problem, the *minimum connected dominating set* (CDS). Finding the minimum CDS is known to be NP-hard, but efficient approximate solutions exist which give satisfactory results (Guha and Khuller, 1998).

3.3. Matching Multiple Images Simultaneously

So far, all methods have started from *pairwise* image matching. Multi-view matches are obtained afterwards by transitive linking. There are also methods that directly recover matches across multiple images. Perhaps the first work explicitly designed for the challenges of (at the time much smaller) unordered image sets was (Schaffalitzky and Zisserman, 2002). Descriptors from *all* images are stored in one joint (binary) space-partitioning tree, so that one can efficiently query sets of descriptors that all lie within some similarity threshold. Having the descriptors of the entire dataset in one search structure avoids pairwise matching and instead directly delivers a set of putative multi-view matches in linear time, but that set is heavily contaminated with false matches. The putative matches between each pair of images are then counted to obtain an approximate match graph. Finally, the edges of the graph are pruned to a maximum spanning tree.

More recently, direct multi-view matching has been demonstrated across thousands of images (Havlena and Schindler, 2014), again drawing on ideas from vocabulary-based image retrieval. Instead of matching the images in the dataset against each other, they are matched against a very fine visual vocabulary (16 million words, Mikulik et al., 2013). Matching is thus reduced to feature quantization, under the additional condition that a visual word may appear at most once in an image

(otherwise the corresponding interest points are discarded, but this concerns only a tiny fraction of the points). Since the set of matches from the vocabulary to all input images by transitivity form a multi-view correspondence, the method is again linear in the number of images, and can be trivially parallelized. Like before, the multi-view matches must nevertheless be broken down into pairwise correspondences (i.e., a match graph), because pose estimation is normally done incrementally by chaining single-view or two-view orientations that can be estimated robustly with RANSAC-type sampling methods.

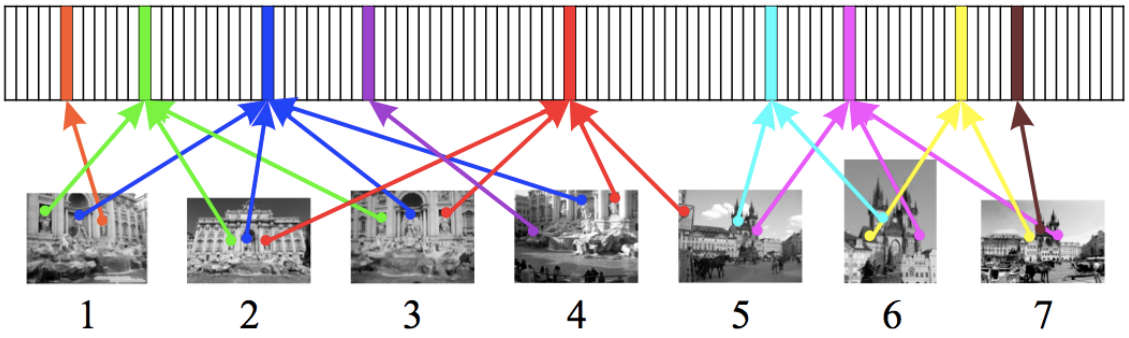

Figure 4: With a sufficiently fine visual vocabulary, quantizing interest point descriptors directly yields unambiguous multi-view correspondences.

4. CONCLUSIONS

We have presented a systematic survey of tie-point matching methods for large, unordered image sets. While the matching of sparse interest points can be considered solved for images with moderate viewpoint changes, large-scale applications face the problem that applying it to all features across all possible image pairs is intractable. There are two levels of speed-ups: on the one hand, one can – for any (ordered or unordered) image set – reduce the time needed to establish tie-point correspondences between two images, by using appropriate data structures and by cleverly selecting the interest points that are used for matching. On the other hand, in unordered image sets the key to efficient matching is to limit the number of times the explicit descriptor matching procedure is executed. This can be

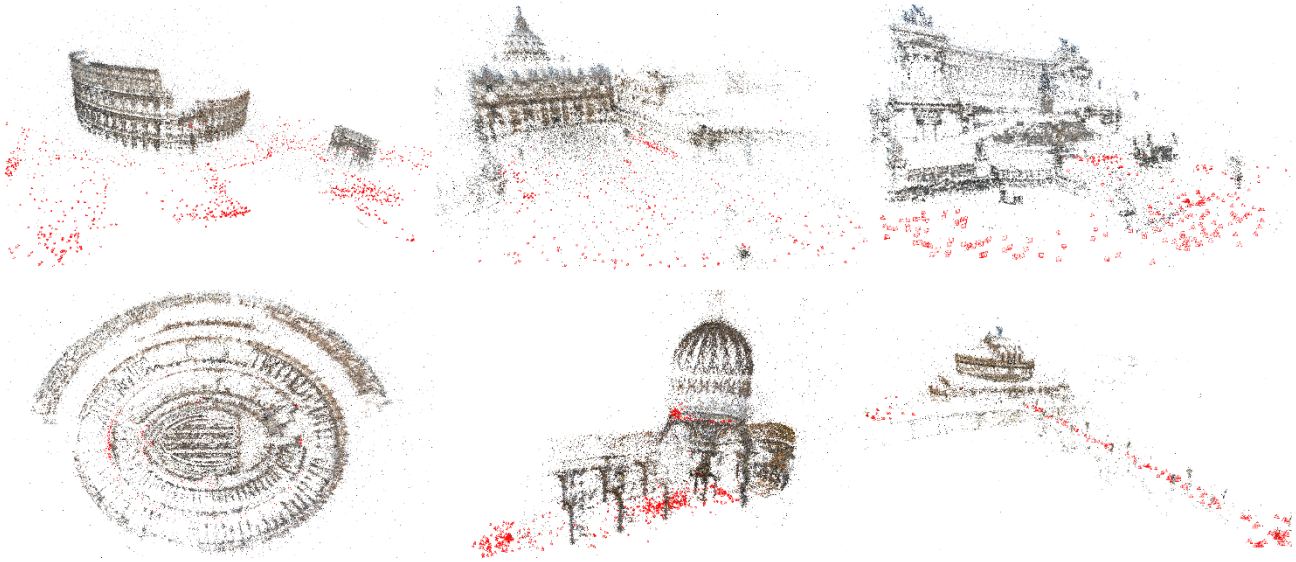

Figure 5: Examples of models found and reconstructed in a dataset of 13'049 crowd-sourced tourist images from Rome. Top row: Colosseum and Constantine Arch (1'392 images), St. Peter's Square (967), Altare della Patria (498). Bottom row: inside of Colosseum (774), inside of St. Peter's Basilica (728), Castel Sant'Angelo (251).

done by adequately decimating the input image set, by focusing on promising image pairs, or by establishing multi-view correspondence directly, rather than separately testing a (quadratically growing) portion of all image pairs. With a well-designed combination of these strategies, it is now possible to generate tie-points across thousands of unordered images in a matter of hours – see Figure 5.

5. REFERENCES

Agarwal, S., Snavely, N., Simon, I., Seitz, S. & Szeliski, R. (2009): Building Rome in a day. International Conference on Computer Vision (ICCV), pp. 72-79.

Frahm, J.-M., Georgel, P., Gallup, D., Johnson, T., Raguram, R., Wu, C., Jen, Y.-H., Dunn, E., Clipp, B., Lazebnik, S. & Pollefeys, M. (2010): Building Rome on a cloudless day. European Conference on Computer Vision (ECCV), pp. 368-381.

Guha, S. & Khuller, S. (1998): Approximation algorithms for connected dominating sets. Algorithmica 20 (4), pp. 374-387.

Hartmann, W., Havlena, M. & Schindler, K. (2014): Predicting matchability. Computer Vision and Pattern Recognition (CVPR), pp. 9-16.

Havlena, M. & Schindler, K. (2014): VocMatch: Efficient multiview correspondence for structure from motion. European Conference on Computer Vision (ECCV), pp. 46-60.

Havlena, M., Torii, A., Knopp, J. & Pajdla, T. (2009): Randomized structure from motion based on atomic 3D models from camera triplets. Computer Vision and Pattern Recognition (CVPR), pp. 2874-2881.

Havlena, M., Torii, A. & Pajdla, T. (2010): Efficient structure from motion by graph optimization. European Conference on Computer Vision (ECCV), pp. 100-113.

Heinly, J., Schönberger, J. L., Dunn, E. & Frahm, J.-M. (2015): Reconstructing the World in Six Days (As Captured by the Yahoo 100 Million Image Dataset). Computer Vision and Pattern Recognition (CVPR), pp. 3287-3295.

Hiep, V., Keriven, R., Labatut, P. & Pons, J.-P. (2009): Towards high- resolution large-scale multi-view stereo. Computer Vision and Pattern Recognition (CVPR), pp. 1430-1437.

Hirschmüller, H. (2008): Stereo processing by semiglobal matching and mutual information. IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (2), pp. 328-341.

Jancosek, M. & Pajdla, T. (2011): Multi-view reconstruction preserving weakly-supported surfaces. Computer Vision and Pattern Recognition (CVPR), pp. 3121-3128.

Li, X., Wu, C., Zach, C., Lazebnik, S. & Frahm, J.-M. (2008): Modeling and recognition of landmark image collections using iconic scene graphs. European Conference on Computer Vision (ECCV), pp. 427-440.

Lowe, D. (2004): Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60 (2), pp. 91-110.

Mikulik, A., Perdoch, M., Chum, O. & Matas, J. (2013): Learning vocabularies over a fine quantization. International Journal of Computer Vision 103 (1), pp. 163-175.

Nistér, D. & Stewénius, H. (2006): Scalable recognition with a vocabulary tree. Computer Vision and Pattern Recognition (CVPR), pp. 2161-2168.

Oliva, A. & Torralba, A. (2001): Modeling the shape of the scene: A holistic representation of the spatial envelope. International Journal of Computer Vision 42 (3), pp. 145-175.

Schaffalitzky, F. & Zisserman, A. (2002): Multi-view matching for unordered image sets, or "How Do I Organize My Holiday Snaps?". European Conference on Computer Vision (ECCV), pp. 414-431.

Sivic, J. & Zisserman, A. (2003): Video Google: A text retrieval approach to object matching in videos. International Conference on Computer Vision (ICCV), pp. 1470-1477.

Snavely, N., Seitz, S. & Szeliski, R. (2008): Skeletal graphs for efficient structure from motion. Computer Vision and Pattern Recognition (CVPR), pp. 1-8.

Wu, C. (2013): Towards linear-time incremental structure from motion. 3D International Conference on 3D Vision (3DV), pp. 127-134.