Guided Matching Based on Statistical Optical Flow for Fast and Robust Correspondence Analysis

Josef Maier[1(✉)], Martin Humenberger[1], Markus Murschitz[1], Oliver Zendel[1], and Markus Vincze[2]

[1] AIT Austrian Institute of Technology, Vienna, Austria
{josef.maier.fl,martin.humenberger,
markus.murschitz.fl,oliver.zendel}@ait.ac.at
[2] Vienna University of Technology, Vienna, Austria
vincze@acin.tuwien.ac.at

Abstract. In this paper, we present a novel algorithm for reliable and fast feature matching. Inspired by recent efforts in optimizing the matching process using geometric and statistical properties, we developed an approach which constrains the search space by utilizing spatial statistics from a small subset of matched and filtered correspondences. We call this method Guided Matching based on Statistical Optical Flow (GMbSOF). To ensure broad applicability, our approach works on high dimensional descriptors like SIFT but also on binary descriptors like FREAK. To evaluate our algorithm, we developed a novel method for determining ground truth matches, including true negatives, using spatial ground truth information of well known datasets. Therefore, we evaluate not only with precision and recall but also with accuracy and fall-out. We compare our approach in detail to several relevant state-of-the-art algorithms using these metrics. Our experiments show that our method outperforms all other tested solutions in terms of processing time while retaining a comparable level of matching quality.

Keywords: Image matching · Correspondence analysis · Statistical optical flow · Guided matching · Ground truth for feature matching

1 Introduction

Many modern real-time computer vision applications, such as visual odometry for autonomous driving or navigation of unmanned aerial vehicles, require not

This work was funded by the Austrian Research Promotion Agency (FFG) project RoSSATA (contract #849035).

Electronic supplementary material The online version of this chapter (doi:10.1007/978-3-319-46478-7_7) contains supplementary material, which is available to authorized users.

B. Leibe et al. (Eds.): ECCV 2016, Part VII, LNCS 9911, pp. 101–117, 2016.
DOI: 10.1007/978-3-319-46478-7_7

only accurate but also fast detection and tracking of distinctive parts across several images. A well known approach to tackle this challenge is called feature matching. A feature is represented by a keypoint and its descriptor, thus, feature matching consists of keypoint detection (*e.g.* FAST [1]), descriptor extraction (*e.g.* SIFT [2] or FREAK [3]), and correspondence analysis. While similarity of descriptors is the main measure for correspondence analysis, higher speed as well as more robustness can be achieved by employing additional information such as statistical distributions of keypoints, estimated geometry, or even a priori knowledge about the scene. Impressive results have been achieved in the past two decades and a summary is given in Sect. 2. However, matching quality and especially processing speed can still be improved to broaden applicability.

Thus, the first contribution of this work is a novel algorithm for fast and robust correspondence analysis (Sect. 3). We call it Guided Matching based on Statistical Optical flow (GMbSOF). The main idea is to constrain the search space by estimating spatial statistics from a small subset of matched and filtered correspondences. It significantly speeds up the matching process compared to state-of-the-art algorithms while maintaining their matching quality.

As a second contribution, we introduce a method to calculate ground truth data for matching that includes true negatives (Sect. 4.1). This data is generated from spatial ground truth information, such as optical flow, disparity, or homographies, provided by publicly available datasets [4–6].

To evaluate our algorithm, we present a detailed comparison with state-of-the-art matching methods (Sects. 4.2 and 4.3) in terms of quality and processing time. Using true negatives and true positives, we are able to compute accuracy $ACC = (TP + TN) / (P + N)$ and fall-out $FPR = FP / (FP + TN)$ in addition to precision and recall. These measures are important, as accuracy enables to quantify the closeness of a matching algorithm's output to the true solution, while fall-out is a direct measure on the algorithm's failure rate in correlation with non-matchable keypoints (true negatives). Therefore, we present – for the first time – accuracy and fall-out values for all compared algorithms.

2 Related Work

We focus on two categories of matching approaches most relevant to the presented work: pure similarity-based techniques and algorithms that additionally use geometrical or statistical keypoint information. We take efficiency as well as matching quality into account.

An interesting approach is the randomized KD-tree [7], which is an approximate nearest neighbor (NN) search algorithm. It works best on SIFT-like descriptors [2] and builds multiple randomized KD-trees which are searched in parallel in order to speed-up the search process. Higher precision is achieved by the slower priority search k-means tree [8]. It clusters data points using the full distance across all dimensions of the descriptors instead of partitioning the data on one dimension at a time. Another fast matching algorithm is CasHash, recently

introduced by Cheng *et al.* [9]. The authors claim that their cascade hashing strategy accelerates matching tenfold or more compared to KD-tree based algorithms. The speed-up is achieved by a three-layer design (lookup, remapping, and ranking) which uses an adopted version of the Locality Sensitive Hashing (LSH) algorithm [10] to generate binary code for hashing.

These algorithms are fast for high dimensional features[1], but they are outperformed by most matching algorithms for binary features. This is because the Hamming distance is used as cost function (descriptor distance) which can be calculated very efficiently [11] compared to the standard L2-norm for high dimensional features. A popular binary feature descriptor is Fast Retina Keypoint (FREAK) [3], which is inspired by the human retina. Alahi *et al.* [3] propose a cascade matching strategy for FREAK which allows to eliminate wrong matches in several steps by comparing only a few bytes of the descriptors (saccadic search). Strecha *et al.* [12] propose a method called LDAHash to convert high dimensional descriptors like SIFT to binary features for speed-up. The hierarchical clustering tree approach of Muja *et al.* [13] works with binary as well as high dimensional features by performing a decomposition of the search space to construct a tree structure.

An attractive geometry-based approach is presented by Shah *et al.* [14]. It first extracts 20 % of SIFT-features with the largest scale and matches them using a KD-tree followed by the ratio test introduced by Lowe [2]. Second, it estimates the fundamental matrix and searches for corresponding features along the epipolar lines. Unfortunately, this approach only works for images without dominant planes.[2] Hu *et al.* [16] use SIFT descriptor similarity in addition to the distance between matching keypoints to start an iterative voting-scheme based on the PageRank algorithm [17]. Torki and Elgammal [18] suggest a graph matching scheme. They embed all features within an Euclidean space where their locations reflect both, the descriptor similarity and the spatial arrangement. They match multiple feature sets by solving an Eigen-value problem and achieve linear complexity compared to the typical quadratic problem complexity of other graph matching methods.

In addition, several approaches exist that use multiple homographies of small segmented areas between the images in order to constrain the search space [19–22]. After similarity based matching followed by a filtering step, homographies are calculated from scale and orientation of the corresponding SIFT features. These homographies are further transferred into Hough space where voting is performed. The resulting homographies are used to filter the matches. These approaches lead to good precision and recall but the execution takes several seconds. An early approach of using multiple homographies was presented by Jung and Lacroix [23]. This algorithm relies on randomly finding an initial homography between local groups of their own features. Therefore, cross

[1] Note that binary descriptors can also be high dimensional. However, we refer to descriptors for which their distance is calculated using the L2-norm.

[2] This issue could be resolved by utilizing the method of Chum *et al.* [15] instead of the 8-point-algorithm.

correlation on the pixel intensity in addition to a similarity measure on affine transformation parameters and on the cornerness is used. This initial homography further guides the matching process. Features near already established matches in the first image are chosen. This process is repeated on local groups of features and the homography is updated. Huo *et al.* [24] estimate one global homography to constrain the matching process. They first downsample the image pairs, match SIFT features, and perform a ratio test. Based on these matches, a homography is estimated and adopted for image pairs at high resolution, where guided matching is performed. This technique can be applied on data with single planar surfaces only.

Geiger *et al.* [25] developed a matching framework for visual odometry which uses their own features and descriptors in addition to statistical information from pre-matched features. In a first pass, only a subset of the features is extracted, matched, and filtered using cross check and an application of the delaunay triangulation [26]. Next, the features of the first image are assigned to cells of an equally spaced grid. For each cell the minimum and maximum displacement of its feature set is calculated. These statistics are further used to constrain the final search space, which speeds up the matching. It is closely related to our method because it also uses statistics on the displacement of features to guide the matching. Mill's [27] method also uses statistics for filtering SIFT matches. After matching with a KD-tree, histograms on the change of orientation and scale of SIFT features are used in combination with Lowe's ratio test to reject all matches that do not belong to the three central bins. This hybrid filtering approach leads to higher precision than with ratio test alone. Unfortunately no recall was given. Sun *et al.* [28] incorporate a Gaussian Mixture Model (GMM) similar to Chui and Rangarajan [29] in the matching process by considering both, feature similarity and spatial information. They model a point set using the GMM and assign each GMM component a different weight given by the feature similarity. Thus, they achieve increased robustness for scenes with high outlier ratio. Unfortunately, only the number of correct matches instead of precision, recall, and processing time is given in their paper.

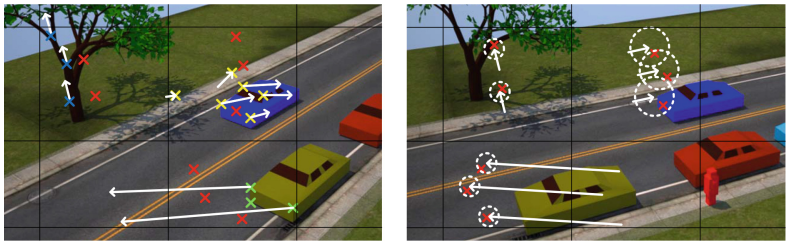
Additionally to the matching algorithms above, we want to highlight important post-processing methods. A very interesting approach is called Vector Field Consensus (VFC) [30,31] and relies on the calculation of an interpolated flow field based on correspondences generated by any kind of feature matcher. To achieve this, it assigns a mixture model with the assumption that the noise is Gaussian for inliers and uniform for outliers. The parameters of the mixture model are estimated using the EM algorithm [32]. Moreover, the method uses a smoothness criterion on the vector field for stabilization which leads to higher robustness but also rejects true positives at flow discontinuities originating from, *e.g.*, borders in the scene. They also suggested a cascade scheme of their algorithm [33] by applying the algorithm on matched features filtered by the ratio test in the first place. Second, n nearest neighbors are considered using the parameters from the previous execution for initialization of VFC. This yields to a significantly increased number of true positives. Different variants of their

algorithm also accept various geometric models (*e.g.* fundamental matrix or homography) [34] which can be used for filtering. In addition to the impressive results on precision and recall, their implementation only needs a few milliseconds for filtering, which is negligible compared to the processing time of most matching algorithms. Lin *et al.* [35] presented a similar filtering approach that applies the smoothness criterion, *e.g.*, on the likelihood of the motion (called bilateral motion field) instead of on the motion itself. Thus, enabling motion discontinuities at object boundaries. This leads to a decreased rejection of true positive matches. The estimated bilateral motion field can further be used to robustly expand the set of matches. In addition to these mentioned post-processing methods, several RANSAC-based methods exist (*e.g.* [36–43]) which use various geometric models for filtering the matches.

Many of the mentioned algorithms either suffer from high processing time ([16,19–23,28,29]), rely on geometric assumptions of the environment ([14,24]), or are limited to specific descriptors and keypoint types ([3,8–10,12,16,19–23,25,27]). In the following sections, we present a novel approach to overcome these limitations. We support both, high dimensional and binary descriptors.

3 Guided Matching Based on Statistical Optical Flow

The goal of our algorithm is to significantly reduce the search space for feature matching. We use displacement information from a subset of feature matches to accelerate the matching process without relying on any assumptions. This speeds up the matching process while achieving quality measures comparable to state-of-the-art algorithms. It consists of four steps, where Fig. 1 illustrates the process. First, we find the most distinctive features (blue, green, and yellow crosses in



(a) Left image (b) Right image

Fig. 1. Overview of the GMbSOF algorithm (best viewed in color). (a) Feature subsampling (yields to blue, green, and yellow crosses) & initial matches (white arrows). (b) SOF & guided matching: The remaining keypoints of the left image (red crosses) are mapped to the right image using SOF (white arrows) where the corresponding right keypoints are searched within a small search space indicated by the dashed circles. Due to the large flow differences within the cell containing the blue (moving) car, the search radius is enlarged compared to the others, where the flow is rather constant. (The color figure can be found online)

Fig. 1(a)) in both images by local non-maxima suppression of their responses (Sect. 3.1). This allows fast similarity based matching (second step, Sect. 3.1) on only a few features distributed over the whole image. The arrows in Fig. 1(a) represent the optical flow vectors resulting from these initial matches. Third, we calculate the SOF for the initial matches and fourth, we use SOF (arrows and dashed circles in Fig. 1(b)) to guide the matching of all remaining features, which are shown as red crosses in Fig. 1 (Sects. 3.2 and 3.3). The following sections explain these steps in detail.

3.1 Feature Selection and Initial Matching

The number of features is reduced by using the responses (*e.g.* blob strength of SIFT or corner strength of FAST [1]) of the keypoints. This is effective, but applying a fixed threshold based on the global response range across the whole image is not favorable. Depending on the scene, keypoints with high local but low global response might be deleted, whereas others with a low local but high global response might be kept. Both cases would decrease the matching quality. To solve this problem, we use a scheme that finds a proper threshold within a certain neighborhood by analyzing local response differences. To define a neighborhood, we divide the image into a regular grid, where the number of cells depends on the image dimensions. The aim is to keep the number of features within a cell (depending on the scene) in the same range, independent of the image size. Next, the maximum response difference $\Delta r_j = \hat{r}_j - \check{r}_j$ is calculated for each cell j, where $\hat{r}_j$ is its maximum and $\check{r}_j$ its minimum keypoint response. In addition, $\Delta r_{j,i} = \hat{r}_j - r_{j,i}$ is computed for each keypoint i within a cell. We accept all keypoints that satisfy $\Delta r_{j,i} \leq a\Delta r_j$ with $a < 1$. If the responses are not equally distributed within Δr_j, a too large number of keypoints would be accepted. Therefore, we iteratively decrease[3] a to lower the effective threshold $a\Delta r_j$ and, thus, the number of accepted keypoints while keeping the strongest.

Finally, similarity based matching using a hierarchical clustering tree [13] for binary features or a randomized KD-tree [7] for high dimensional features followed by a ratio test with a threshold of 0.75 is performed on this subset.

3.2 Statistical Optical Flow

SOF is used to guide the matching process, thus, to estimate an initial search position in the second image and to reduce the search range to a small area. It consists of statistics about the spatial displacements of the initial matches and is independently and sparsely estimated for several areas of the image. To do this, we use another regular grid, based on the spatial distribution of the initial matches. The cell size z is calculated in a way that the average number of initial matches in each cell k is large enough for a meaningful statistic[4]. As

[3] Each third of accepted keypoints halves a, which is initialized with $a = 0.25$.

[4] In our experiments, 16 turned out to be a well balanced compromise between runtime and robustness. Naturally, adapting this number depending on environment, camera configuration, and feature type, could lead to better results.

only the average number of matches per cell is used to determine z, some cells may not contain enough matches. As a result, matches from neighboring cells are added until the minimum number of matches $(u_{k,l}\ v_{k,l}) \leftrightarrow \left(u'_{k,l}\ v'_{k,l}\right)$ is reached. In the next step, statistics over the magnitudes and angles[5] of the flow vectors from the matches with index l are calculated for each cell k. Therefore, the flow vectors $\mathbf{f}_{k,l} = (\Delta u_{k,l}\ \Delta v_{k,l})^T$ are calculated with $\Delta u_{k,l} = u'_{k,l} - u_{k,l}$ and $\Delta v_{k,l} = v'_{k,l} - v_{k,l}$. From vectors $\mathbf{f}_{k,l}$ the distances $d_{k,l} = \|\mathbf{f}_{k,l}\|$ and angles $\alpha_{k,l} = \angle(\mathbf{f}_{k,l})$ are determined. For all $d_{k,l}$ and $\alpha_{k,l}$, the mean values $\bar{d}_k$ and $\bar{\alpha}_k$, in addition to the median values $\tilde{d}_k$ and $\tilde{\alpha}_k$ are calculated for every cell k. The statistics vector $\mathbf{q}_k = \left(\bar{d}_k\ \tilde{d}_k\ \bar{\alpha}_k\ \tilde{\alpha}_k\right)^T$ of a cell is accepted (or valid) if the condition

$$accept_{\boxdot}(\mathbf{q}_k): \quad \left|\frac{\bar{d}_k - \tilde{d}_k}{\bar{d}_k}\right| \leq b \ \boxdot\ \left|\frac{\bar{\alpha}_k - \tilde{\alpha}_k}{\bar{\alpha}_k}\right| \leq b, \ \boxdot \in \{\vee, \wedge\} \tag{1}$$

holds. For this filtering step, $accept_{\vee}(\mathbf{q}_k)$ in most cases leads to better results of SOF than $accept_{\wedge}(\mathbf{q}_k)$. We traced this back to two reasons: First, if some keypoint positions within a cell underlie small localization errors, this can result in validation values exceeding the threshold b for either the distance or the angles, but typically not both. In contrast, the probability of false matches to share the same angle or the same distance is very low in most cases. The second reason for $accept_{\vee}(\mathbf{q}_k)$ are cells with objects at varying depths, which for many camera configurations mainly result in varying distances but similar angles. If (1) does not hold, $\mathbf{q}_k$ is considered invalid and, thus, rejected. During our analysis, a value of $b = 0.3$ led to good results. However, it can be improved if b is adapted in a range of $[0.3, 0.75]$ according to the inlier ratio. As the true inlier ratio is not known, we estimate an *inlier ratio tendency factor* $\varphi_e = n_e/n_o$ given by the number of matches n_e, which survived the ratio test after initial matching, and the number n_o of left-frame keypoints after response-filtering. φ_e is linearly proportional to the true inlier ratio as can be seen in Fig. 2(a). Thus, b is adapted linearly by $b = \beta^{S,F}\varphi_e + b_0^{S,F}$ in the previously mentioned range where different parameters are used for binary (β^F, b_0^F) and high dimensional features (β^S, b_0^S). From the accepted statistic vectors (1) an overall statistic is calculated using the median values $\tilde{d}_k$ and $\tilde{\alpha}_k$. Their mean $\bar{d}_{\tilde{d}}$ and $\bar{\alpha}_{\tilde{\alpha}}$ in addition to the standard deviation $\sigma_{\tilde{d}}$ and $\sigma_{\tilde{\alpha}}$ are used to calculate the threshold values

$$T_{1,2}^{d,\alpha} = \left[\bar{d}_{\tilde{d}} \pm c\sigma_{\tilde{d}}\ \ \bar{\alpha}_{\tilde{\alpha}} \pm c\sigma_{\tilde{\alpha}}\right] \tag{2}$$

to filter distances $d_{k,l}$ and angles $\alpha_{k,l}$. For this threshold estimation we set the parameter $c = 4$ to exclude far outliers. As in this filtering step some distances and angles are removed from their cells, the number of values $d_{k,l}$ and $\alpha_{k,l}$ might be below the minimum number requested for each cell. We compensate

[5] For statistics about the angles, we correct them in a way that the angular distances within the whole set are $\leq \pi$.

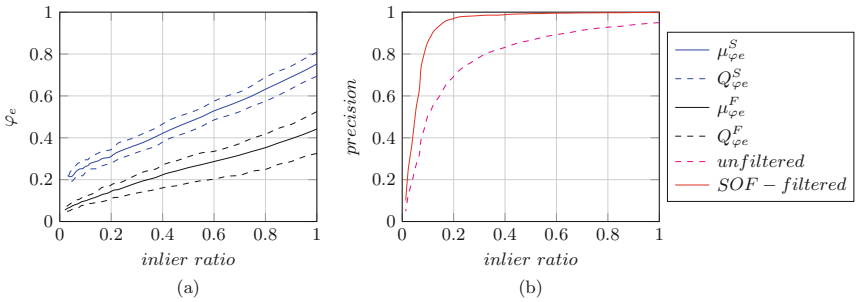


Fig. 2. (a) Varying inlier ratio *vs.* average inlier ratio tendency factors $\mu^S_{\varphi_e}$ and $\mu^F_{\varphi_e}$ in addition to their quartiles $Q^S_{\varphi_e}$ and $Q^F_{\varphi_e}$ for the entire KITTI flow dataset [4] using SIFT features $(\mu^S_{\varphi_e}, Q^S_{\varphi_e})$ as well as FAST keypoints in conjunction with FREAK descriptors $(\mu^F_{\varphi_e}, Q^F_{\varphi_e})$. (b) Precision after initial matching before and after SOF estimation & filtering step using the KITTI disparity dataset [4] in conjunction with FAST keypoints & FREAK descriptors. The true inlier ratio was generated synthetically using our evaluation framework described in Sect. 4.1. Additional results can be found in the supplementary material.

this by adding values from neighboring cells. Using these values, vectors $\mathbf{q}_k$ are recalculated for each cell and validated using $accept_\wedge(\mathbf{q}_k)$ of (1) to be more restrictive. Next, the standard deviations σ^d_k and σ^α_k are calculated from all $d_{k,l}$ and $\alpha_{k,l}$ for every valid cell k separately. Each σ^d_k is used to estimate the search radius

$$s_k = c\sigma^d_k \tag{3}$$

for subsequent guided matching. Since small values for c lead to decreased matching quality and high values to increased processing time, as can be seen in Fig. 3, we found in $c = 3.5$ a well-balanced compromise.

Each $\mathbf{q}_k$ marked as invalid (according to $accept_\wedge(\mathbf{q}_k)$) is replaced by the most similar vector $\mathbf{q}_{k_s}$ of all valid neighbors and the overall (over every $d_{k,l}$ and $\alpha_{k,l}$) statistic. The search radius s_k of such a cell is enlarged utilizing $\mathbf{q}_k$ of the invalid cell and $\mathbf{q}_{k_s}$. Thus, the standard deviation σ^d_k is replaced by

$$\sigma^d_k = \sigma^d_{k_s} + \frac{\left\|\tilde{\mathbf{f}}_k - \tilde{\mathbf{f}}_{k_s}\right\|}{c} \text{ where} \tag{4}$$

$$\tilde{\mathbf{f}}_i = \left(\tilde{d}_i cos\left(\tilde{\alpha}_i\right) \;\; \tilde{d}_i sin\left(\tilde{\alpha}_i\right)\right)^T, \; i \in \{k, k_s\}. \tag{5}$$

Subsequently, the search range for invalid cells is calculated with (3) and the standard deviation $\sigma^\alpha_k = m\sigma^\alpha_{k_s}$ with $m = min\left(max\left(\sigma^\alpha_k/\sigma^\alpha_{k_s},\ 1\right), 1.5\right)$.

Next, the statistical optical flow $\boldsymbol{F} = \left[\bar{\mathbf{f}}_1 \; \bar{\mathbf{f}}_2 \; \cdots \; \bar{\mathbf{f}}_{n_b}\right]^T$ with vectors $\bar{\mathbf{f}}_k = \left(\bar{d}_k cos\left(\bar{\alpha}_k\right) \;\; \bar{d}_k sin\left(\bar{\alpha}_k\right)\right)^T$ and the total number of cells n_b is estimated. Besides, the initial matches are filtered using (2) with σ^d_k and σ^α_k in addition to $\bar{d}_k$ and $\bar{\alpha}_k$. Thus, we are able to constrain the search space in the second image using $\boldsymbol{F}$ and $\mathbf{s} = (s_1 \; s_2 \; \cdots \; s_{n_b})^T$.

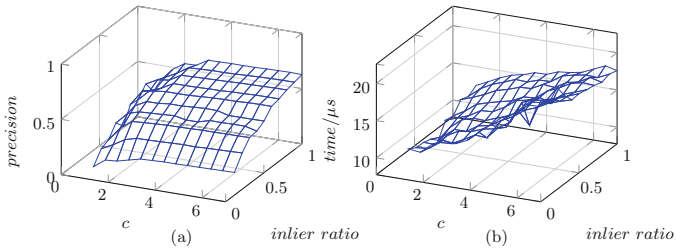


Fig. 3. Influence of parameter c on the matching quality and runtime performed on the entire "wall" dataset [5,6] for SIFT features. (a) Mean precision and (b) mean runtime for matching one feature over various inlier ratios as well as parameter values of c for estimating the final search range s_k.

To reduce border effects between cells, we divide each cell k into a sub-grid with a size of 5×5. Within this sub-grid, the SOF values of the inner 3×3 cells remain the same as in the original grid, whereas the outer cells are linearly interpolated using their neighbors of the original grid. The search radii of the outer cells are enlarged to cover the entire search space of the surrounding cells. Finally, $\boldsymbol{F}$, $\mathbf{s}$, n_b, and the cell size z in pixels are replaced by the interpolated SOF, which is used during the guided matching process.

As shown in Fig. 2(b), the quality of SOF significantly decreases for inlier ratios below 20 %. Analysis of all tested datasets showed, that this corresponds to $\varphi_e < 0.2$ for high dimensional features and $\varphi_e < 0.08$ for binary features. Thus, we use the inlier tendency factor φ_e to decide whether or not SOF should be estimated and used. If SOF is not used, slower similarity-based matching is performed instead of guided matching on the remaining features using a hierarchical clustering tree [13] for binary features and a randomized KD-tree [7] for high dimensional features, followed by a ratio test. We apply the VFC algorithm on these results and accept its output only if more than 10 % of the input matches survive the filtering step. Otherwise, the matches after the ratio test are accepted without further filtering.

3.3 Guided Matching

After estimation of SOF, the remaining keypoints are matched. We use $\boldsymbol{F}$ to map left keypoint locations $\mathbf{x}_i = (u_i \; v_i)^T$ to the right image. Then, for each $\mathbf{x}_i$ a descriptor-similarity-based ranking of matching right image keypoints $\mathbf{x}'_{i,m}$ within a certain search radius is done (Algorithm 1). For better readability, the cell indices k of SOF are replaced by x and y which specify the position of a cell inside the grid. In addition to the search radii $\mathbf{s}$, we define the minimum search radius r_{min}[6].

After matching, we perform a ratio test which obviously can only be performed if two nearest neighbors are available. Thus, if only one corresponding

[6] Throughout our evaluations we set $r_{min} = 10$, but a higher value should be considered if there are small non-rigid elements present in the scene for which their initial correspondences might be filtered out during the SOF estimation.

Algorithm 1. Guided matching

for $i \leftarrow 1$, # of left keypoints **do**
  Calculate SOF grid position $(x\ y)^T = \left(\left\lfloor \frac{u_i}{z} \right\rfloor \left\lfloor \frac{v_i}{z} \right\rfloor\right)^T$
  Get search position $\tilde{\mathbf{x}}'_i = \mathbf{x}_i + \boldsymbol{F}_{x,y}$
  KD-tree spatial search radius $r_{kd} = max\left(\mathbf{s}_{x,y},\ r_{min}\right)$
  Search keypoints $\mathbf{x}'_{i,m}$ with $\left\|\tilde{\mathbf{x}}'_i - \mathbf{x}'_{i,m}\right\| < r_{kd}$
  for all $\mathbf{x}'_{i,m}$ found **do**
    Calculate descriptor similarity
  Sort keypoints based on similarity in ascending order

keypoint is found, we perform a crosscheck. Finally, if this returns more than one possible match, we perform the ratio test. Otherwise, the found keypoint must be within 66 % of its corresponding search radius $\mathbf{s}_{x,y}$ to be accepted.

4 Experimental Results

Most authors performing tests on their matching algorithms or comparing different approaches only use precision and recall, which are in fact expressive quality measures. However, in our opinion accuracy and fall-out are evenly important and should not be neglected. To be able to calculate accuracy and fall-out, the true negatives have to be known. Thus, we developed a framework (Sect. 4.1) which is able to generate ground truth matches out of spatial ground truth information. This is the only input for our evaluations, no additional image data is used. We compare the processing time (Sect. 4.2) of 8 different algorithms. Additionally, we evaluate statistics on accuracy, fall-out, precision, and recall dependent on varying inlier ratios (from 1 % to 100 %) for each dataset-algorithm-combination (Sect. 4.3 and supplementary material) (additional results can be found in the supplementary material). For all evaluations, exactly the same parameters were used for GMbSOF which were found by performing tests with varying parameters on all datasets.

We use datasets KITTI flow and disparity [4] as well as all possible image pair combinations of "bark", "boat", "graffity", and "wall" from Mikolajczyk *et al.* [5,6]. All of them provide spatial ground truth which was used for generation of ground truth matches and, thus, for evaluation of the different methods. As the datasets provided by KITTI are quite sparse, due to the fact that they were created using laser range data, a pre-processing step was necessary to allow a meaningful evaluation. It fills as many invalid ground truth pixels (where no laser range data was available) as possible using information of the neighboring pixels. For this, we apply a carefully designed local median filter variant which preserves discontinuities and avoids filtering artifacts.

We compare our algorithm (GM) to selected state-of-the-art algorithms, namely CasHash (CH) [9], hierarchical clustering tree (HC) [13], priority search k-means tree (HK) [8], SparseVFC (VFC) [30,31] in combination with the hierarchical clustering tree, linear matching (LI) and Locality Sensitive Hashing (LSH) [44] from the FLANN library [8], as well as the randomized KD-tree

(RA) [7].[7] Even if the method of Geiger *et al.* [25] and the geometry-aware feature matching method of Shah *et al.* [14] are relevant, we had to exclude them from our evaluations. A comparison with [25] would lead to distorted results due to necessary adaption in order to use SIFT or FREAK. Geometry-aware feature matching [14] failed for most disparity image pairs.[8]

4.1 The Evaluation Framework

In this section, we use designations left and right images also for flow datasets (left refers to the first image and right to the second). For calculation of ground truth matches and evaluation of matching results, we first limit the possible matching keypoints by two proximity constraints: (i) Every keypoint in the left image can be mapped to a unique position in the right image by employing the available spacial ground truth. (ii) There must be at least one keypoint within the close proximity (see below) of the mapped position. If these constraints are fulfilled, we first treat all right keypoints with a maximal displacement error of 5 % of the upper bound of the spacial ground truth magnitudes as reasonable. To be more robust, we ignore keypoints corresponding to the upper 20 % of the distances to the mapped positions. Then, the median distance $\tilde{d}_n$ and its median absolute deviation $\tilde{\sigma}_n$ are calculated for the remaining 80 %. Distances larger than $\tilde{d}_n + 3.5\tilde{\sigma}_n$ are rejected. The highest remaining distance equals the radius t_d that encircles the most reasonable candidates.

Now we address the core problem of finding unambiguous one-to-one matches within those most reasonable candidates. A match is considered as unambiguous and valid if:

1. A keypoint in the left image is within t_d in the right image,
2. The similarity of their descriptors is below a threshold (160 for 512bit binary descriptors) and smaller than 1.5 times the similarity of the second best match within t_d,
3. 1 and 2 also hold from right to left.

Then, the smallest similarity d_{min} of all remaining matches within t_d is computed and all correspondences that have a similarity exceeding $1.25 \times d_{min}$ are rejected.

True negative matches (missing corresponding keypoint in the left or right image), which are necessary to calculate accuracy and fall-out, emerge in the first place from left keypoints for which no reasonable matching candidates are found. Additional true negatives are found in both images during the reduction of ambiguity by filtering the matches.

To reach a specific inlier ratio, we first equalize the number of keypoints in both images by randomly deleting true negatives. Next, we randomly remove

[7] In this context, we have to mention that not all of the above algorithms could be tested on all different keypoint and descriptor types as some algorithms accept only one specific type.

[8] We used the unchanged code provided by the authors.

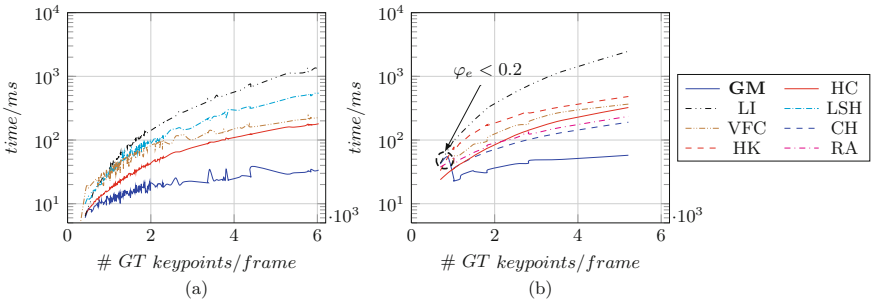


Fig. 4. Runtime analysis (see footnote 9) of different matching algorithms (a) on the KITTI flow dataset [4] using FAST keypoints & FREAK descriptors, (b) on the "wall" dataset from Mikolajczyk *et al.* [5,6] using SIFT features. For the allocation of abbreviations to their full names please see Sect. 4. Each datapoint stems from a different image pair. Additional results can be found in the supplementary material.

keypoints (matching or non-matching) alternating from both images until the desired inlier ratio is reached.

We use the same keypoint type for ground truth generation and matching. However, for the filtering steps we always use FREAK, because it is fast. There is no notable difference in terms of quality of the ground truth matches when using FREAK instead of SIFT descriptors. Only the number of matches changes slightly. The quality of the resulting matches depends on the spatial ground truth quality. Ground truth quality deficencies can in most cases be compensated by the presented keypoint filtering approach. We performed randomized visual inspection of the ground truth matches for all datasets. There were no false matches for the KITTI databases and only one for the considered Mikolajczyk datasets (due to false spatial ground truth at this image location).

4.2 Runtime Analysis

We measure the runtime[9] of the above mentioned algorithms for each image pair of the entire KITTI disparity, flow [4] and the "wall" [5,6] datasets separately. Then the runtime with respect to the number of input keypoints is analyzed. For all datasets, an inlier ratio of 75 % was used.[10] As can be seen in Fig. 4, our algorithm (GM) outperforms all tested state-of-the-art algorithms in terms of runtime. Especially compared to the randomized KD-tree and the CasHash algorithm (Fig. 4(b)), which are among the fastest matching algorithms for high dimensional features, our approach is approximately 3.5–4.0 times faster for around 5000 matches. An even higher improvement is achieved for binary features (see Fig. 4(a)).

[9] Time measurements were performed using the smallest runtime of 100 runs on an Intel Xeon E5-2687W 3.1 GHz CPU.

[10] For this inlier ratio, the number of keypoints (true positives and true negatives) is close to its maximum for most datasets which allows a better performance analysis.

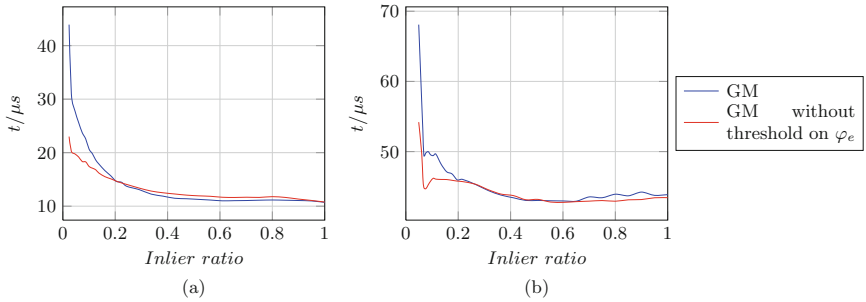


Fig. 5. Varying inlier ratio compared to the average matching time (see footnote 9) per keypoint (over the whole algorithm) using (a) FAST keypoints & FREAK descriptors and (b) SIFT features for the KITTI flow dataset from Menze and Geiger [4]. This evaluation was performed on the entire dataset, keeping the number of left and right keypoints equal for all inlier ratios and each image pair separately. For comparison, our algorithm was evaluated with and without switching to a similarity-based matcher for low inlier ratio tendency factors φ_e.

The spikes of the runtime marked by a dashed black circle in Fig. 4(b) originate from switching to the fall-back similarity-based matching instead of guided matching, as the inlier ratio tendency factor φ_e was below its threshold (0.2 for high dimensional features). The fall-back solution is not triggered by the low number of features but by the difficulty of the evaluated scene.

The approach is evidently the most scalable with respect to the number of features, since for a high number of features the processing time only slightly increases. To investigate the dependency of our algorithm's processing time on SOF accuracy, we perform additional runtime evaluations for varying inlier ratios (see Fig. 5). The runtime remains nearly constant until the inlier ratio decreases below 20 %–40 %. For inlier ratios below 10 %, the probability of φ_e to fall below the specified threshold value for switching to a similarity-based matcher raises quickly. In that case, the runtime is determined by the used similarity-based matching algorithms.

4.3 Matching Quality Evaluations

To asses the matching quality in comparison to the other algorithms, we evaluate the common quality measures precision and recall, but also accuracy and fall-out. The mean values of those quality measures are shown in Fig. 6[11]. In order to perform a fair comparison with our algorithm, a ratio test was performed on the results of each matching algorithm.

Comparing our algorithm to the others, we observe a significantly higher recall, and a slightly higher fall-out. These characteristic differences can consistently be observed for all datasets (examples are shown in Fig. 6(d)–(f)). Both

[11] Additional results can be found in the supplementary material.

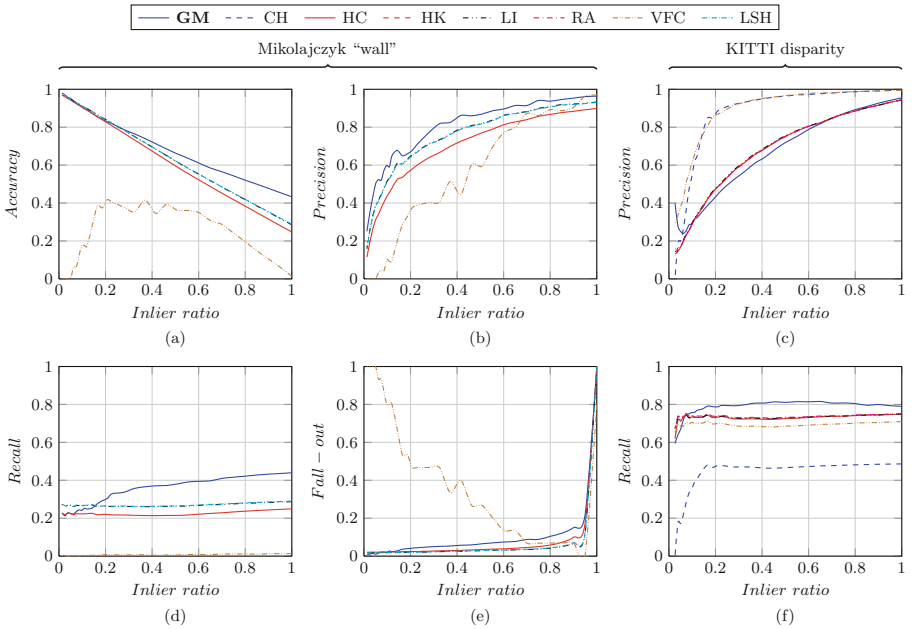


Fig. 6. Varying inlier ratio compared to mean (a) accuracy, (b) precision, (d) recall, and (e) fall-out for the "wall" dataset from Mikolajczyk *et al.* [5,6] using FAST keypoints & FREAK descriptors. Moreover, the mean (c) precision and (f) recall are shown for the KITTI disparity dataset [4] using SIFT features. For abbreviations see Sect. 4.

differences can be traced back to the very small search space during the guided matching process originating from the estimated SOF. Our solution, in contrast to other algorithms, is more robust against repetitive or similar patterns as long as they appear only once within our search radii **s** and it is possible to find initial matches sufficiently distributed over the whole image. This yields higher recall.

5 Conclusion

In this paper we presented our novel approach for highly efficient and fast feature matching, called Guided Matching based on Statistical Optical Flow (GMbSOF). It estimates the search space using statistics for certain areas of the image determined by a subset of matched features. Using these, we constrain the search space which dramatically accelerates the matching process. In most publications, matching algorithms are only tested using precision and recall, which are in fact very meaningful. However, in our opinion accuracy and fall-out are also very important for classification of a matching algorithm. To compute these quality measures, true negatives are required. Therefore, we developed a framework to determine true negatives and ground truth matches from datasets providing spatial ground truth information. A comprehensive comparison with

relevant state-of-the-art algorithms showed that our method outperforms all of them in terms of processing time while achieving comparable matching quality. The limitation of our approach is the reliable detection of very small dynamic objects, as it is likely that they are filtered out during the calculation of the statistical optical flow.

References

1. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 430–443. Springer, Heidelberg (2006). doi:10.1007/11744023_34
2. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
3. Alahi, A., Ortiz, R., Vandergheynst, P.: FREAK: Fast Retina Keypoint. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 510–517 (2012)
4. Menze, M., Geiger, A.: Object scene flow for autonomous vehicles. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3061–3070 (2015)
5. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Trans. Pattern Anal. Mach. Intell. **27**(10), 1615–1630 (2005)
6. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Gool, L.V.: A comparison of affine region detectors. Int. J. Comput. Vis. **65**(1–2), 43–72 (2005)
7. Silpa-Anan, C., Hartley, R.: Optimised KD-trees for fast image descriptor matching. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–8 (2008)
8. Muja, M., Lowe, D.G.: Scalable nearest neighbor algorithms for high dimensional data. IEEE Trans. Pattern Anal. Mach. Intell. **36**(11), 2227–2240 (2014)
9. Cheng, J., Leng, C., Wu, J., Cui, H., Lu, H.: Fast and accurate image matching with cascade hashing for 3D reconstruction. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–8 (2014)
10. Charikar, M.S.: Similarity estimation techniques from rounding algorithms. In: Proceedings of the Thiry-Fourth Annual ACM Symposium on Theory of Computing, STOC 2002, pp. 380–388. ACM (2002)
11. Zinner, C., Humenberger, M., Ambrosch, K., Kubinger, W.: An optimized software-based implementation of a census-based stereo matching algorithm. In: Bebis, G., et al. (eds.) ISVC 2008. LNCS, vol. 5358, pp. 216–227. Springer, Heidelberg (2008). doi:10.1007/978-3-540-89639-5_21
12. Strecha, C., Bronstein, A.M., Bronstein, M.M., Fua, P.: LDAHash: improved matching with smaller descriptors. IEEE Trans. Pattern Anal. Mach. Intell. **34**(1), 66–78 (2012)
13. Muja, M., Lowe, D.G.: Fast matching of binary features. In: Proceedings of the 9th Conference on Computer and Robot Vision, CRV 2012, pp. 404–410. IEEE Computer Society (2012)
14. Shah, R., Srivastava, V., Narayanan, P.J.: Geometry-aware feature matching for structure from motion applications. In: IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 278–285 (2015)
15. Chum, O., Werner, T., Matas, J.: Two-view geometry estimation unaffected by a dominant plane. In: IEEE Conference on Comput. Vision and Pattern Recognition (CVPR), vol. 1, pp. 772–779 (2005)

16. Hu, M., Liu, Y., Fan, Y.: Robust image feature point matching based on structural distance. In: Tan, T., Ruan, Q., Wang, S., Ma, H., Di, K. (eds.) IGTA 2015. CCIS, vol. 525, pp. 142–149. Springer, Heidelberg (2015). doi:10.1007/978-3-662-47791-5_17
17. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: bringing order to the web. Technical report 1999–66, Stanford InfoLab (1999)
18. Torki, M., Elgammal, A.: One-shot multi-set non-rigid feature-spatial matching. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3058–3065 (2010)
19. Chen, H.Y., Lin, Y.Y., Chen, B.Y.: Co-segmentation guided hough transform for robust feature matching. IEEE Trans. Pattern Anal. Mach. Intell. **37**(12), 2388–2401 (2015)
20. Chen, H.Y., Lin, Y.Y., Chen, B.Y.: Robust feature matching with alternate hough and inverted hough transforms. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2762–2769 (2013)
21. Puerto-Souza, G.A., Mariottini, G.L.: A fast and accurate feature-matching algorithm for minimally-invasive endoscopic images. IEEE Trans. Med. Imaging **32**(7), 1201–1214 (2013)
22. Puerto-Souza, G.A., Mariottini, G.L.: Hierarchical Multi-Affine (HMA) algorithm for fast and accurate feature matching in minimally-invasive surgical images. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2007–2012 (2012)
23. Jung, I.K., Lacroix, S.: A robust interest points matching algorithm. In: IEEE International Conference on Computer and Vision, vol. 2, pp. 538–543 (2001)
24. Huo, C., Pan, C., Huo, L., Zhou, Z.: Multilevel SIFT matching for large-size VHR image registration. IEEE Geosci. Remote Sens. Lett. **9**(2), 171–175 (2012)
25. Geiger, A., Ziegler, J., Stiller, C.: StereoScan: dense 3D reconstruction in real-time. In: IEEE Intelligent Vehicles Symposium, pp. 963–968 (2011)
26. Shewchuk, J.R.: Triangle: engineering a 2D quality mesh generator and Delaunay triangulator. In: Lin, M.C., Manocha, D. (eds.) WACG 1996. LNCS, vol. 1148, pp. 203–222. Springer, Heidelberg (1996). doi:10.1007/BFb0014497
27. Mills, S.: Relative orientation and scale for improved feature matching. In: 20th IEEE International Conference on Image Processing (ICIP), pp. 3484–3488 (2013)
28. Sun, K., Li, P., Tao, W., Liu, L.: Point sets matching by feature-aware mixture point matching algorithm. In: Tai, X.-C., Bae, E., Chan, T.F., Lysaker, M. (eds.) EMMCVPR 2015. LNCS, vol. 8932, pp. 392–405. Springer, Heidelberg (2015). doi:10.1007/978-3-319-14612-6_29
29. Chui, H., Rangarajan, A.: A feature registration framework using mixture models. In: IEEE Workshop on Mathematical Methods in Biomedical Image Analysis, pp. 190–197 (2000)
30. Ma, J., Zhao, J., Tian, J., Yuille, A.L., Tu, Z.: Robust point matching via vector field consensus. IEEE Trans. Image Process. **23**(4), 1706–1721 (2014)
31. Ma, J., Zhou, H., Zhao, J., Gao, Y., Jiang, J., Tian, J.: Robust feature matching for remote sensing image registration via locally linear transforming. IEEE Trans. Geosci. Remote Sens. **53**(12), 6469–6481 (2015)
32. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. J. Royal Stat. Soc. **39**(B), 1–38 (1977)
33. Ma, J., Ma, Y., Zhao, J., Tian, J.: Image feature matching via progressive vector field consensus. IEEE Sig. Process. Lett. **22**(6), 767–771 (2015)

34. Ma, J., Qiu, W., Zhao, J., Ma, Y., Yuille, A.L., Tu, Z.: Robust L_2E estimation of transformation for non-rigid registration. IEEE Trans. Signal Process. **63**(5), 1115–1129 (2015)
35. Lin, W.-Y.D., Cheng, M.-M., Lu, J., Yang, H., Do, M.N., Torr, P.: Bilateral functions for global motion modeling. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8692, pp. 341–356. Springer, Heidelberg (2014). doi:10.1007/978-3-319-10593-2_23
36. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (1981)
37. Torr, P.H.S., Zisserman, A.: MLESAC: a new robust estimator with application to estimating image geometry. Comput. Vis. Image Underst. **78**(1), 138–156 (2000)
38. Chum, O., Matas, J., Kittler, J.: Locally optimized RANSAC. In: Michaelis, B., Krell, G. (eds.) DAGM 2003. LNCS, vol. 2781, pp. 236–243. Springer, Heidelberg (2003). doi:10.1007/978-3-540-45243-0_31
39. Nistér, D.: Preemptive RANSAC for live structure and motion estimation. In: International Conference on Computer Vision (ICCV), pp. 199–206 (2003)
40. Chum, O., Matas, J.: Matching with PROSAC - progressive sample consensus. In: Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, pp. 220–226 (2005)
41. Raguram, R., Frahm, J.-M., Pollefeys, M.: A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008. LNCS, vol. 5303, pp. 500–513. Springer, Heidelberg (2008). doi:10.1007/978-3-540-88688-4_37
42. Chum, O., Matas, J.: Optimal randomized RANSAC. IEEE Trans. Pattern Anal. Mach. Intell. **30**(8), 1472–1482 (2008)
43. Ni, K., Jin, H., Dellaert, F.: GroupSAC: efficient consensus in the presence of groupings. In: International Conference on Computer Vision (ICCV), pp. 2193–2200 (2009)
44. Andoni, A., Indyk, P.: Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. Commun. ACM **51**(1), 117–122 (2008)