

# ISE

Industrial and  
Systems Engineering

## Boosting RANSAC Via Dual Principal Component Pursuit

YUNCHEN YANG<sup>1</sup>, XINYUE ZHANG<sup>1</sup>, TIANJIAO DING<sup>1</sup>, DANIEL P. ROBINSON<sup>2</sup>,  
RENÉ VIDAL<sup>3</sup>, AND MANOLIS C. TSAKIRIS<sup>1</sup>

<sup>1</sup>Information Science and Technology, Shanghai Tech University

<sup>2</sup>Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, USA

<sup>3</sup>Fellow, IEEE

ISE Technical Report 22T-016



LEHIGH  
UNIVERSITY.

# Boosting RANSAC via Dual Principal Component Pursuit

Yunchen Yang<sup>1</sup> Xinyue Zhang<sup>1</sup> Tianjiao Ding<sup>1</sup> Daniel P. Robinson<sup>2</sup> René Vidal<sup>3</sup> Manolis C. Tsakiris<sup>1</sup>

## Abstract

In this paper, we revisit the problem of local optimization in RANSAC. Once a so-far-the-best model has been found, we refine it via Dual Principal Component Pursuit (DPCP), a robust subspace learning method with strong theoretical support and efficient algorithms. The proposed DPCP-RANSAC has far fewer parameters than existing methods and is scalable. Experiments on estimating two-view homographies, fundamental and essential matrices, and three-view homographic tensors using large-scale datasets show that our approach consistently has higher accuracy than state-of-the-art alternatives.

## 1 Introduction

An important problem in 3D vision is to fit subspaces to observations corrupted by noise and outliers, e.g. when fitting planes to point clouds, or estimating homographies, fundamental matrices or homographic tensors from feature correspondences detected and matched from images. One of the most widely used robust subspace estimation schemes in the past few decades is Random Sample Consensus (RANSAC) [7], which iteratively computes a model from a random sample of data points, scores this candidate model by measuring its consensus with all data points, and keeps it if it has the highest consensus score. The iterations stop once the algorithm reaches a confidence  $p$  that an outlier-free sample has been found. In the noiseless case, the expected number of iterations  $T$  to reach such confidence  $0 \leq p \leq 1$  can be approximated by

$$T \approx \frac{\log(1-p)}{\log(1-(1-e)^{N_s})}, \quad (1)$$

where  $e$  is the fraction of outliers and  $N_s$  is the number of points in each sample needed to compute a model. Here, the assumption is that the model computed from an outlier-free sample is close to the ground-truth model.

In practice, when the data are noisy, it has been shown that RANSAC may need many more than  $T$  iterations to obtain a subspace model that is close to the ground-truth

one [4]. The reason is that due to noise not all outlier-free samples give an equally accurate estimate, thus violating the aforementioned assumption. Inspired by this, [4] proposed to refine the model each time RANSAC finds a model that has a consensus score higher than all previous iterations, termed *so-far-the-best* model. This procedure, known as *local optimization* or LO-RANSAC, includes an extra inner RANSAC step that further samples from inliers to the so-far-the-best model, and recursively applies direct linear transform (DLT) with decreasing thresholds. LO-RANSAC has been shown to be more accurate and require fewer iterations, albeit it is computationally more demanding [11]. To lighten the computational burden, [11] proposed to use a fraction of the inliers to the so-far-the-best model in the local optimization step. Nevertheless, the whole local optimization pipeline still requires many parameters, such as the number of inner RANSAC iterations and the range and step-size for the threshold decrease, and is therefore ad-hoc and complex. Recently, [1] proposed an alternative local optimization procedure that uses tools from graph theory. It defines an energy function based on spatial coherence and distance to the model, and applies the graph-cut algorithm to obtain a separation of inliers and outliers. This is an alternative to the usual way of labeling points by hard-thresholding the distance to the model. However, the refitting tends not to be robust since the graph-cut often gives false positives.

On the other hand, many robust subspace learning methods based on sparse and low-rank representation have appeared in the last decade. While such methods benefit from both strong theoretical guarantees and efficient algorithms, most of them are based on low-rank assumptions, hence they cannot be applied to robust estimation tasks in geometric vision, which require estimating a hyperplane. One exception is Dual Principal Component Pursuit (DPCP) [25, 27, 30, 6, 5], which is suitable for learning subspaces of dimension close to the ambient dimension, e.g. hyperplanes. Remarkably, DPCP has been shown to provably tolerate  $M = \mathcal{O}(N^2)$  outliers, where  $N$  and  $M$  are the numbers of inliers and outliers respectively, under some uniformity assumptions on the data distribution. This is in sharp contrast to the typical guarantee  $M = \mathcal{O}(N)$  of alternative methods. Moreover, many algorithms have been proposed for DPCP, including recursive linear pro-

gramming [25, 27], alternating minimization [17, 27], iteratively reweighted least squares (DPCP-IRLS) [26, 27, 13], projected subgradient (DPCP-PSGM) [30, 6] or Riemannian subgradient [29, 14] methods, and geodesic gradient descent [16]. Notably, DPCP-IRLS and DPCP-PSGM have been shown to outperform the vanilla RANSAC in comparative studies that employ the same running time budget for the task of estimating 3D road planes [30, 6], homography matrices and tensors [5], as well as trifocal tensors [27].

In this paper we use tools from robust subspace learning to address noise and outliers simultaneously in the local optimization step of RANSAC. In particular, we propose to refine the so-far-the-best model via DPCP. Our local optimization scheme has fewer parameters than existing methods and is efficient. Experiments on estimating two-view homographies, fundamental matrices, and three-view homographic tensors using state-of-the-art datasets show that our approach consistently achieves higher accuracy than alternatives such as Graph-Cut RANSAC [1], with comparable running time.

## 2 A review of local optimization in RANSAC

As mentioned above, even if a model is estimated from an all-inlier sample, it may significantly deviate from the true model due to the presence of noise. This is particularly the case when one uses minimal samples, e.g. 3 points for estimating a plane or 7 for a fundamental matrix. Thus, the key idea of local optimization is to find more inliers based on the current model, and use them to compute a better estimate. Since this procedure induces additional computational cost, it is typically done only when a so-far-the-best model has been found<sup>1</sup>. We summarize the locally optimized RANSAC in Algorithm 1, then review two important lines of work in proposing algorithms for line 9 as follows.

### 2.1 The classical local optimization

One way to find more inliers is to look at points that are close to the so-far-the-best model  $M$ . Indeed, [4] proposes to take all points that are within  $\epsilon$  distance to  $M$ , and run DLT on them. However, there may not be a single  $\epsilon$  that is always optimal. The reason is that a good  $\epsilon$  depends not only on the noise distribution, but possibly also on how much  $M$  deviates from the true model. For example, when  $M$  is mildly off from the true model, one would expect to use a larger  $\epsilon$  to include more inliers. On the other hand, a too-large  $\epsilon$  would result in outliers being included, which DLT is not robust to. To resolve those challenges, [4] further proposes a two-fold approach. It first enlarges the threshold to  $K\epsilon$  with  $K > 1$ , and runs DLT. Then, it shrinks  $K$

<sup>1</sup>Hence, local optimization is expected to execute at most  $\log(T) + 1$  times [4], where  $T$  is given in (1).

---

#### Algorithm 1: Locally optimized RANSAC

---

```

1 Input: data points  $P$ ; threshold  $\epsilon$ ; max iteration  $T_{max}$ ; confidence  $p$ ;
2 Output: best model  $M^*$ ;
3 Initialization:  $k \leftarrow 0$ ;  $r^* \leftarrow -\infty$ ;  $T \leftarrow T_{max}$ ;
4 while  $k < T$  and  $k < T_{max}$  do
5    $S_k \leftarrow$  draw a minimal sample from  $P$ ;
6    $M_k \leftarrow$  estimate a model from  $S_k$ ;
7    $r_k \leftarrow$  score  $M_k$  on  $P$  with  $\epsilon$ ;
8   if  $r_k > r^*$  then
9      $M^*, r^* \leftarrow$  local optimization( $M_k, P, \epsilon$ );
10     $T \leftarrow$  expected number of iterations needed to reach confidence  $p$ ;
11  end
12   $k \leftarrow k + 1$ ;
13 end
14 return  $M^*$ 

```

---

and repeats the process until  $K$  drops to 1. To be robust to outliers, this process is further embedded into an extra inner RANSAC step, which takes non-minimal samples from inliers to  $M$ . Overall, this method gives more accurate estimates than RANSAC, but it is computationally demanding. For example, [11] shows that it could reduce the error in pixels by 78% compared to plain RANSAC, while it could use an order of magnitude larger running time<sup>2</sup>. Moreover, this outlier rejection process requires many parameters (§1), and the iterative threshold decreasing process is adhoc. For a summary of the above approach, see [11, Algorithm 2, 3].

### 2.2 The graph-cut local optimization

---

#### Algorithm 2: Local Optimization Pipeline

---

```

1 Input: so-far-the-best model  $M$ ; data points  $P$ ; threshold  $\epsilon$ ; max LO iterations  $k_{max}$ ;
2 Output: best model  $M^*$ ; best score  $r^*$ ;
3 Initialization:  $k \leftarrow 0$ ;  $r^* \leftarrow -\infty$ ;
4 while  $k < k_{max}$  do
5    $I \leftarrow$  classify inliers to  $M$  from  $P$  with  $\epsilon$ ;
6    $M \leftarrow$  estimate a model from  $I$ ;
7    $r \leftarrow$  score  $M$  on  $P$  with  $\epsilon$ ;
8   break if  $r \leq r^*$ ;
9    $r^* \leftarrow r$ ;  $M^* \leftarrow M$ ;  $k \leftarrow k + 1$ ;
10 end
11 return  $M^*, r^*$ 

```

---

Recently, [1] argues that the classical approach is ad-

<sup>2</sup>Note that [11] proposed to use a fraction of the inliers with the so-far-the-best model in the local optimization step to reduce computation.

hoc and complex, and proposes a new local optimization pipeline (Algorithm 2). Based on that pipeline, it further proposes a graph-theoretical approach for line 5. It designs an energy function that considers both the spatial coherence of data points  $P$  and distances from  $P$  to the model  $M$ , which is optimized using graph-cut to obtain a separation of inliers and outliers. Then, a DLT is performed on the inliers  $I$  to  $M$  in order to fit a new model  $M$ . Experimentally, [1] shows that the composition of the pipeline and graph-cut, termed Graph-Cut-RANSAC, achieves higher accuracy and comparable running time to [4, 11].

### 3 DPCP for geometric vision

We have seen that the challenge in refining a model is to be robust to both noise and outliers. In this section, we review **Dual Principal Component Pursuit** (DPCP) [25, 27, 30, 6, 5], a robust subspace learning method that can be applied to various geometric vision problems. Given a dataset  $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_N]$ , where  $\mathbf{y}_i \in \mathbb{R}^D$  either lies close to a hyperplane  $S$  or is an arbitrary outlier in  $\mathbb{R}^D$ , DPCP estimates a normal vector  $\mathbf{b}$  to  $S$  by minimizing the sum of the distances from the data points to the hyperplane:

$$\min_{\mathbf{b}} \sum_i^N |\mathbf{b}^\top \mathbf{y}_i| \quad \text{s.t.} \quad \|\mathbf{b}\|_2 = 1. \quad (2)$$

It is shown in [25, 27, 30, 6, 5] that, under certain conditions, global minima of (2) are guaranteed to be orthogonal to  $S$  (noiseless data) or close to orthogonal to  $S$  (noisy data).

#### 3.1 Robust fundamental matrix estimation

It is well known that a two-view point-point correspondence  $(\mathbf{x}, \mathbf{x}') \in \mathbb{R}^3 \times \mathbb{R}^3$  associated with fundamental matrix  $\mathbf{F} \in \mathbb{R}^{3 \times 3}$  must satisfy the epipolar constraint  $\mathbf{x}'^\top \mathbf{F} \mathbf{x} = 0$ , or equivalently  $(\mathbf{x} \otimes \mathbf{x}')^\top \mathbf{f} = 0$ , where  $\mathbf{f} := \text{vec}(\mathbf{F})$  is the vectorized fundamental matrix. We refer to  $(\mathbf{x} \otimes \mathbf{x}')$  as the epipolar embedding of  $(\mathbf{x}, \mathbf{x}')$ . Given a set of correspondences  $\{(\mathbf{x}_i, \mathbf{x}'_i)\}_{i=1}^N$  that contains sufficiently many (noisy) inliers and possibly a large fraction of mismatches, we propose to robustly estimate an  $\mathbf{F}$  via

$$\arg \min_{\mathbf{f}} \sum_i^N |\mathbf{f}^\top (\mathbf{x}_i \otimes \mathbf{x}'_i)| \quad \text{s.t.} \quad \|\mathbf{f}\|_2 = 1. \quad (3)$$

On the other hand, when the 3D points are lying on a single plane or the translation between two views is zero, it has been shown in [5] that there is a three-dimensional vector space of fundamental matrices that explain the correspondences, and the true fundamental matrix is one of them. In this case, one could recover an essential matrix as follows. First, use DPCP to robustly estimate a 3-dimensional

nullspace by computing

$$\hat{\mathbf{B}} := \arg \min_{\mathbf{B}} \sum_i^N \|\mathbf{B}^\top (\mathbf{x}_i \otimes \mathbf{x}'_i)\|_2 \quad \text{s.t.} \quad \mathbf{B}^\top \mathbf{B} = \mathbf{I}_3, \quad (4)$$

and then calculate the subspace distance between the embeddings and the nullspace,  $w_i := \|\hat{\mathbf{B}}^\top (\mathbf{x}_i \otimes \mathbf{x}'_i)\|_2$ . Second, weight each epipolar embedding by  $w_i$ , and extract a 5-dimensional principal subspace from the weighted data. This in turn gives 5 linear constraints, which one can use to solve for an essential matrix by further considering the polynomial constraints from the essential variety.

This weighting procedure is inspired by the observation demonstrated in Figure 1(a), where the inlier set identified by a minimal estimate via the 5pt algorithm [22] (line 6, Algorithm 1) with a threshold on Sampson distance is usually corrupted by outliers. We see that the algebraic residual given by the model estimated by DPCP precisely characterizes the distribution of the reprojection errors in Figure 1(b). Moreover, Figure 1(c) and (b) compare the model before (c) and after (d) the local optimization via DPCP.

#### 3.2 Robust homography estimation

A recent work [5] proposed to robustly estimate homographies via DPCP, and shows that it performs on par with USAC [18], while running an order of magnitude faster. We briefly review the formulation for completeness.

Let  $(\mathbf{x}, \mathbf{x}')$  be a two-view point-point correspondence related by homography  $\mathbf{H}$ , i.e.,  $\mathbf{x}' \sim \mathbf{H}\mathbf{x}$ . With  $\mathbf{h} := \text{vec}(\mathbf{H})$ , this relation is equivalent to  $\psi_j(\mathbf{x}, \mathbf{x}')^\top \mathbf{h} = 0$ ,  $j = 1, 2$ , where  $\psi_j$  are the so-called homographic embeddings [5][8, §4.1]. Similar to the case of fundamental matrix estimation, if  $\{(\mathbf{x}_i, \mathbf{x}'_i)\}_{i=1}^N$  contains sufficiently many correspondences that are approximately related by  $\mathbf{H}$  as well as some outliers, one can robustly estimate  $\mathbf{H}$  via DPCP:

$$\min_{\mathbf{h}} \sum_i^N \|\mathbf{h}^\top [\psi_1(\mathbf{x}_i, \mathbf{x}'_i), \psi_2(\mathbf{x}_i, \mathbf{x}'_i)]\|_2 \quad \text{s.t.} \quad \|\mathbf{h}\|_2 = 1. \quad (5)$$

Consider now the case of having *three* views, where a point-point-point correspondence  $(\mathbf{x}, \mathbf{x}', \mathbf{x}'')$  is related by two homographies  $\mathbf{H}, \mathbf{G}$ , i.e.,  $\mathbf{x} \sim \mathbf{H}\mathbf{x}' \sim \mathbf{G}\mathbf{x}''$ . This is captured by what is known as a homographic tensor  $\mathcal{H} \in \mathbb{R}^{3 \times 3 \times 3}$  and the relation  $\phi_j(\mathbf{x}, \mathbf{x}', \mathbf{x}'')^\top \mathbf{h} = 0$ ,  $j = 1, \dots, 7$ , where  $\mathbf{h} := \text{vec}(\mathcal{H})$  [5, 20, 21]. The  $\phi_j$ 's are referred to as three-view homographic embeddings. Again, given  $\{(\mathbf{x}_i, \mathbf{x}'_i, \mathbf{x}''_i)\}_{i=1}^N$  corrupted by noise and outliers, the homographic tensor can be robustly estimated via DPCP:

$$\begin{aligned} \min_{\mathbf{h}} \sum_i^N \|\mathbf{h}^\top [\phi_1(\mathbf{x}_i, \mathbf{x}'_i, \mathbf{x}''_i), \dots, \phi_7(\mathbf{x}_i, \mathbf{x}'_i, \mathbf{x}''_i)]\|_2 \\ \text{s.t.} \quad \|\mathbf{h}\|_2 = 1. \end{aligned} \quad (6)$$



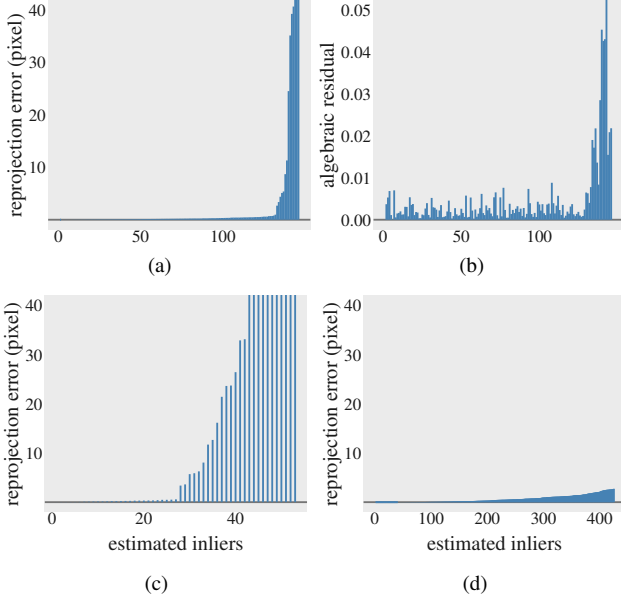


Figure 1: (a) shows the reprojection errors of the inliers estimated by the 5pt algorithm [22] with respect to the ground-truth camera pose. (b) shows the subspace distances from the epipolar embeddings to the model estimated by DPCP (4) corresponding to the ordering in (a). (c) shows the reprojection errors of inliers estimated from another minimal sample with respect to the ground-truth and (d) shows the reprojection errors of the inlier set refined by DPCP based on the model in (c). The local optimization is done by weighting the embeddings by DPCP (4), extracting the 5th principle subspace of the weighted data, and solving for the polynomial constraints of the essential matrix.

## 4 Local optimization with DPCP

In this paper, we propose to apply DPCP in local optimization (Algorithm 2). More precisely, in line 5, we select inliers  $I$  to the current model  $M$  by taking the points from  $P$  that are within  $\epsilon$  distance to  $M$ . Then in line 6, we robustly estimate a model via DPCP from the set  $I$ , as detailed in §3. We refer to this method as DPCP-RANSAC.

One notable advantage of our method is having fewer parameters. Since we use the local optimization pipeline Algorithm 2, there is no need for parameters such as a range and step-size for the threshold decrease as in [4, 11]. Furthermore, while both our method and Graph-Cut-RANSAC [1] are based on Algorithm 2, the latter requires a radius parameter for building a neighborhood graph, and a parameter in the energy for balancing the unary and pairwise terms, which in principle require careful tuning. In contrast, our method does not have parameters in the objective. Therefore, DPCP-RANSAC can be more conveniently applied to various tasks and datasets.

## 5 Experiments

In this section, an experimental study is conducted using realistic datasets on the tasks of estimating two-view homography matrices (§5.3), fundamental (§5.1) and essential matrices (§5.2), and three-view homographic tensors (§5.4).

**Baselines.** We compare the proposed DPCP-RANSAC (R-DPCP) with Graph-Cut-RANSAC (R-GC) [1] and Huber-RANSAC (R-Huber) [19]. For R-GC, we use an off-the-shelf C++ implementation provided by its authors<sup>3</sup>. For a fair comparison we implement R-Huber and R-DPCP in the same C++ framework as R-GC. We also implement a DLT based LO-RANSAC (R-DLT) as a baseline for our comparative study. Although this R-DLT pipeline seems simple, we note that it has not been reported by [4, 11, 19, 1]; we will discuss the differences in §5.1. On the other hand, for the task of homographic tensor estimation, similar C++ implementations as above are not available. Hence, we restrict the comparison for that task in a Matlab environment where we implement R-DLT, R-Huber and R-DPCP. Table 1 summarizes the outlier rejection and non-minimal model refitting steps for the evaluated variants. The confidence  $p$  of these LO-RANSAC variants for all tasks is set to 0.95 with the SPRT test [15, 3] de-activated.

**Experimental setup.** The data points are uniformly sampled during the minimal sampling step (Algorithm 1, line 5). For fairness, we use the same sampling sequence for all RANSAC variants, and vary the sequences for repeated experiments. We employ the truncated quadratic cost [12, 1] as the scoring criterion (Algorithm 1, line 7). For the embedded subspace learning methods, we use the IRLS algorithm to solve for Huber and DPCP, where we set maximum iteration  $\tau_{max} = 100$  and convergence accuracy  $\delta = 10^{-5}$ . Huber has a parameter that controls the combination of quadratic and linear parts, for which we use 0.01 for homography matrix estimation as suggested by [19]. For the remaining tasks, we use  $\{0.1, 0.01, 0.001\}$  and report the mean performance<sup>4</sup>. For Graph-Cut, we use the provided default parameters for each task. In particular, the coherence weight is set to 0.975 and the neighborhood graph is constructed by dividing an image into  $8 \times 8$  grid cells. We note that the description in [1] differs from that in the latest implementation by the authors of [1], and we follow the latest implementation<sup>5</sup>.

<sup>3</sup><https://github.com/danini/graph-cut-ransac>, commit: 0321885.

<sup>4</sup>We note that R-Huber is similar in spirit with R-DPCP, except that R-Huber uses a Huber loss in-lieu of the  $\ell_1$  loss of DPCP. The main disadvantage of Huber is the presence of a parameter to be tuned for different tasks and data. Therefore, for a fair comparison, we report the average over three different values of the Huber parameter. Even though one could tune the parameter for a specific dataset, we empirically observe that the performance of R-Huber always lies between R-DLT and R-DPCP.

<sup>5</sup>Earlier versions would not be suitable for experiments, as later ones contain many improvements and bug fixes.

Table 1: Evaluated LO-RANSAC variants. Outlier rejection and model refitting refer to line 5 and 6 in Algorithm 2.

Algorithm	outlier rejection	model refitting
R-DLT	hard thresholding	DLT-SVD
R-GC	graph-cut	DLT-SVD
R-Huber	hard thresholding	Huber-IRLS
R-DPCP	hard thresholding	DPCP-IRLS

**Metrics.** We use geometric errors as metrics for accuracy, and running time for efficiency. For fundamental and homography matrix estimation, a validation set of inlier correspondences is manually labelled, on which we report the mean Sampson or reprojection error of the estimated matrix. For homographic tensors, we make a comparison in the context of pose estimation. Namely, we take the estimated homographic tensor, decompose it into two pairwise homography matrices, use the provided camera calibration to further reconstruct rotations and translations, and show the angular rotation and angular translation errors with respect to the ground-truth. Since the threshold  $\epsilon$  is often not known a priori and requires tuning in practice, we report geometric errors and running time over different threshold choices to demonstrate the sensitivity of methods. Meanwhile, in each task and dataset, we also tune an optimal threshold for each method, and show the median and interquartile range (IQR) of errors over *all* frames and trials in box plots. The threshold is chosen to correspond to the smallest running time among all thresholds whose error is within 1% of the minimum error.

## 5.1 Fundamental matrix estimation

**Data.** We take 16 image pairs from the Kusvod2<sup>6</sup> dataset and 45 from the EPFL dataset [23]. Kusvod2 contains images from various indoor and outdoor scenes, such as bookshelves, corridors, and buildings, captured by distinct devices. For each pair of images, we use the provided feature correspondences, and the annotated ground-truth inliers as the validation set (see the beginning of §5). Beside this standard benchmark dataset, we also include EPFL, which contains outdoor scenes such as city halls and statues. For each image pair, we detect and match features using SURF [2] with a threshold of 30 and a ratio test of 0.7. Notably, due to the presence of many repetitive patterns and high-resolution images, there is a large number of correspondences and mismatches. EPFL provides ground truth camera matrices, which we use to annotate a set of inliers as the validation set. The correspondences are normalized in each

view [10], and the epipolar embeddings (§3.1) are scaled to have unit  $\ell_2$  norm.

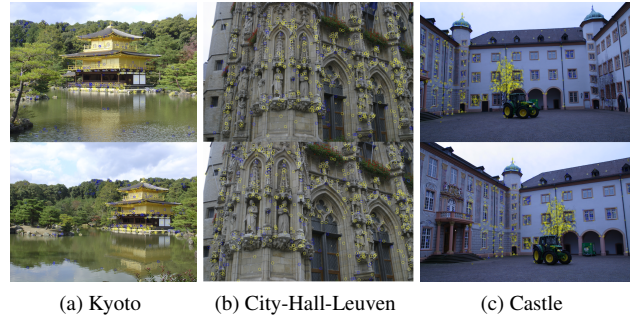


Figure 2: View pairs from Kusvod2 (a) and EPFL (b,c). Correspondences with Sampson distance with respect to the ground-truth less than 1 pixel are marked in yellow (inliers), otherwise in blue (outliers).

**Methods.** The 7-point algorithm [9, 11.1.2] is used as the minimal solver for RANSAC (Algorithm 1, line 6). Since the images are of different sizes, the threshold  $\epsilon$  is set to be a scalar  $\sigma$  multiplied by the diagonal of the image [11], and we run the methods with varying  $\sigma$ .

**Results.** Figure 3 presents the mean Sampson error  $\mathcal{E}_{samp}$  in pixels and running time  $t$  in milliseconds with respect to different threshold multipliers  $\sigma$ , as well as the median and IQR of  $\mathcal{E}_{samp}$  for an optimal  $\sigma$  for each method and dataset. As a first observation, R-DPCP is the method with the smallest Sampson error consistently over all threshold choices, as evidenced by the red curves being the bottom ones in Figure 3. Remarkably, R-DPCP surpasses R-GC [1], a recently proposed LO-RANSAC. For instance, on Kusvod2, R-DPCP reaches a Sampson error of 0.67px with  $\sigma = 10^{-2.6}$ , while R-GC reaches that of 1.00px with the same  $\sigma$ . Interestingly, R-DPCP seems to exhibit better robustness to the choice of threshold multiplier  $\sigma$ , e.g. on EPFL, the Sampson error of R-DPCP stays below 0.29px for  $\sigma \leq 10^{-2.8}$  and increases only gradually after that, in sharp contrast with all other methods. Finally, R-DPCP appears to be scalable to the number of correspondences: On Kusvod2 which has 120 correspondences on average, all the methods have similar running times which are less than 1msec; on EPFL which contain about 2k correspondences, R-DPCP uses significantly smaller running time than R-GC, e.g. 5.55msec compared to 51.16msec with optimally chosen  $\sigma$ . For more detailed efficiency evaluation, please refer to the supplementary material.

It may seem surprising that R-DLT typically reaches a smaller error than R-GC. We note that while the locally optimized RANSAC in [4, 11] uses DLT to fit models from non-minimal samples, it is different from the R-DLT here. Rather than doing iterative DLT with a decreasing thresh-

<sup>6</sup><http://cmp.felk.cvut.cz/data/geometry2view/index.xhtml>

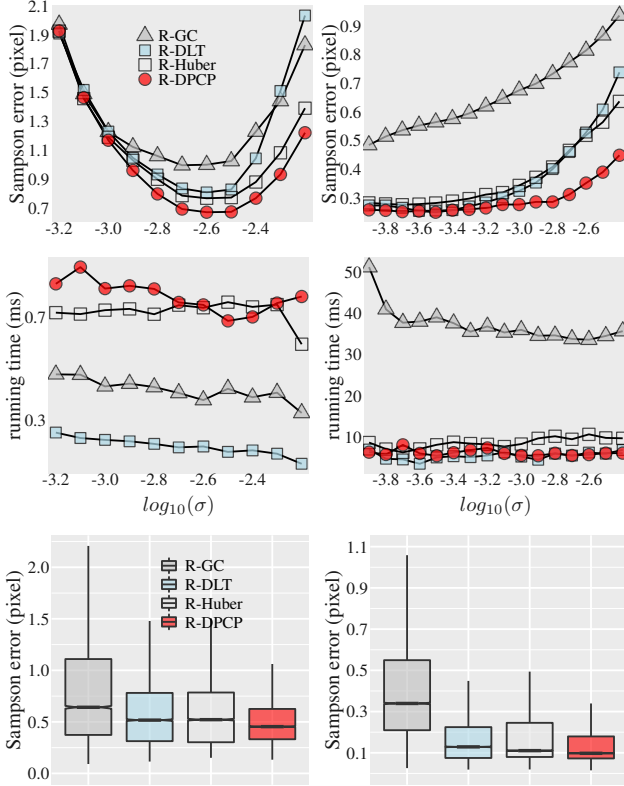


Figure 3: Fundamental matrix estimation using 16 image pairs from Kusvod2 (left) and 45 from EPFL (right). All methods terminate when a confidence of  $p = 95\%$  has been reached. The top two rows show Sampson error in pixels and running time in milliseconds with varying threshold multiplier  $\sigma$ , while the bottom row shows Sampson error with threshold optimally tuned for each method. Experiments are averaged over 500 trials.

old, R-DLT uses a fixed threshold and stops the iteration if the model does not give a higher score, as in line 5 and 8 of Algorithm 2. As such, R-DLT differs from the LO, LO<sup>+</sup> or LO<sup>+</sup> reported in [1].

## 5.2 Essential matrix estimation

**Data.** For the task of essential matrix estimation, we take 276 image pairs from the PhotoTourism<sup>7</sup> dataset, which contains photos of historical sites collected by tourists with very different devices. Each image pair gives around 2000 correspondences, of which roughly 30% are inliers. Interestingly, the image pairs from PhotoTourism are either capturing planar structures or have small to zero translations, as in Figure 4. According to [5], there is a 3-dimensional vector space of fundamental matrices that approximately explains the inlier correspondences, and the true essential

<sup>7</sup><http://phototour.cs.washington.edu/>

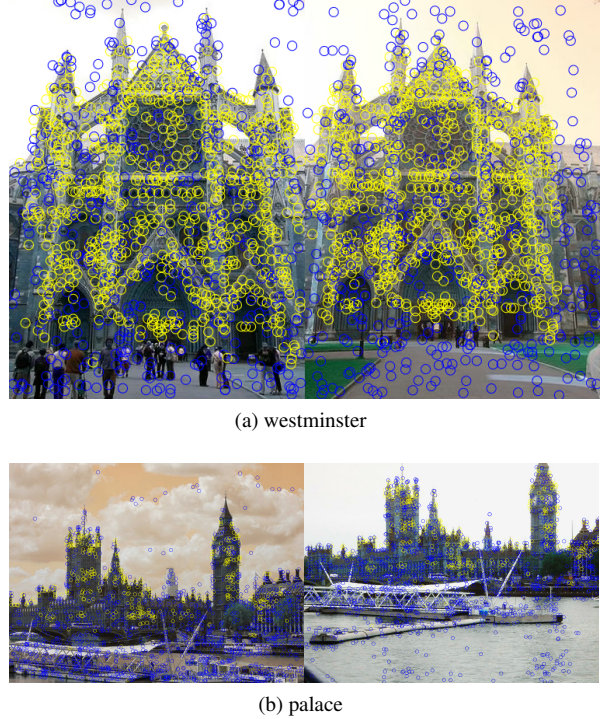


Figure 4: View pairs from PhotoTourism. Correspondences with reprojection error smaller than 5 pixels (inliers) are marked in yellow, otherwise in blue (outliers).

matrix is one of them.

**Methods.** Again, we vary the threshold multiplier  $\sigma$  to control the threshold of Sampson errors as measured in pixels. The minimal sampler takes 5 correspondences as linear constraints intersecting the essential variety and solves a polynomial system with 3 unknown [22]. During the local optimization, we take the 5 principle component of all estimated inliers as the linear constraints and solve the polynomial system with 3 unknowns. However, since the embeddings approximately lie on a lower dimensional subspace, R-DPCP and R-Huber estimates a 3-dimensional nullspace and take the inverse of cosine subspace angle from the nullspace to the epipolar embeddings as weights for the extraction of principal component. Which serves as a outlier rejection strategy via re-weighting.

**Results.** In Figure 5, we report the mean reprojection error  $\mathcal{E}_{repr}$  in pixels and running time  $t$  in milliseconds with respect to different  $\sigma$ , as well as the median and IQR of  $\mathcal{E}_{repr}$  with an optimal  $\sigma$  tuned for each method. Again, we observe that R-DPCP gives the smallest error among all choices of thresholds compared to other methods. Moreover, we note that with a larger threshold multiplier  $\sigma$ , the running time of all methods decreases. This is likely due to the fact that with a larger threshold, more inliers tend to be counted, hence less RANSAC iterations are required to



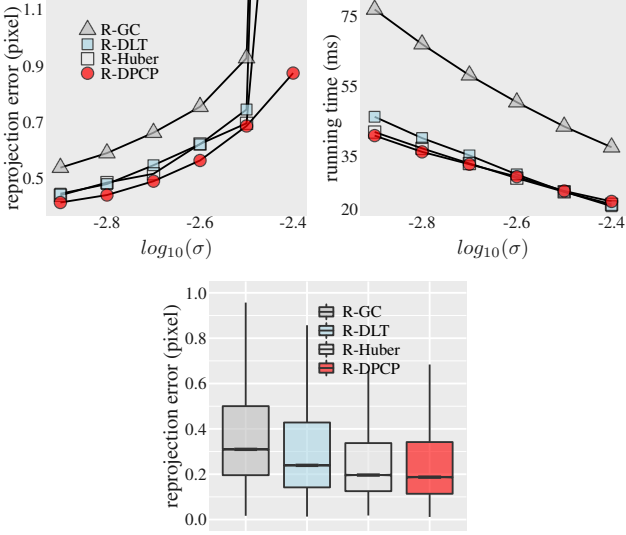


Figure 5: Essential matrix estimation using 276 image pairs from the Phototourism dataset. All methods terminate when a confidence of  $p = 95\%$  has been reached. The top row shows reprojection error and running time with varying threshold multiplier  $\sigma$ , while the bottom row shows reprojection error with threshold optimally tuned for each method. Experiments are averaged over 300 trials. Reprojection error is in pixels and running time is in milliseconds.

reach the confidence  $p$ .

### 5.3 Homography estimation

**Data.** We take 14 image pairs from the Extreme View Dataset (EVD)<sup>8</sup> and 115 from the Hpatches dataset<sup>9</sup>. They collect image pairs of textured planar structures, e.g. graffiti and posters, as well as those with small baselines [28]. For EVD, we use the provided feature correspondences for homography estimation, and the validation set of inliers for evaluation. For Hpatches, we detect and match features using SURF [2] with a threshold of 30 and a ratio test of 0.7. As shown in Figure 6, image pairs from these datasets are challenging for being highly noisy and having extreme view-point changes. We perform a similar normalization to the correspondences and homographic embeddings as described in §5.1.

**Methods.** 4 pairs of correspondences are drawn as minimal samples, from which a homography matrix can be estimated [9]. We vary  $\sigma$  to control the threshold of reprojection errors as measured in pixels.

**Results.** In Figure 7, we report the mean reprojection error  $\mathcal{E}_{repr}$  in pixels and running time  $t$  in milliseconds with

<sup>8</sup><http://cmp.felk.cvut.cz/wbs>. Image pair ‘cafe’ from EVD is not used as the provided correspondences contain less than 5% inliers.

<sup>9</sup><https://hpatches.github.io>

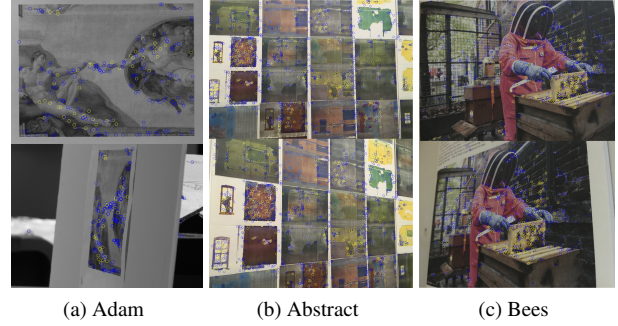


Figure 6: View pairs from EVD (a) and Hpatches (b,c). Correspondences with reprojection error smaller than 5 pixels (inliers) are marked in yellow, otherwise in blue (outliers).

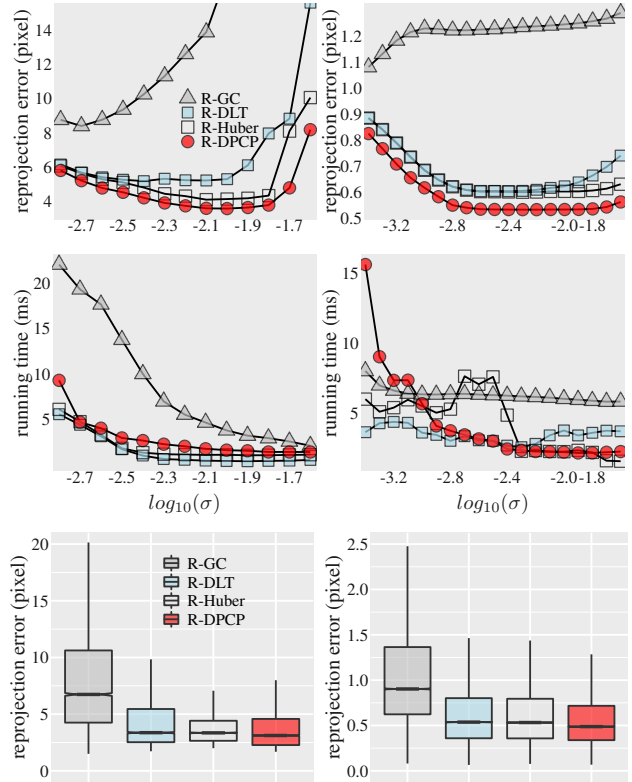
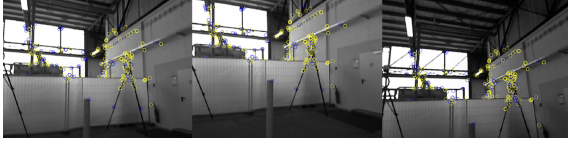


Figure 7: Homography estimation using 14 image pairs from the EVD dataset (left) and 115 from the Hpatches (right). All methods terminate when a confidence of  $p = 95\%$  has been reached. The top two rows show reprojection error and running time with varying threshold multiplier  $\sigma$ , while the bottom row shows reprojection error with threshold optimally tuned for each method. Experiments are averaged over 500 trials. Reprojection error is in pixels, running time is in milliseconds.



(a) Near



(b) Hemi

Figure 8: View triplets from TUM: (a) Near (b) Hemi. Correspondences with reprojection error smaller than 5 pixel are marked in yellow (inliers), otherwise in blue (outliers).

respect to different  $\sigma$ , as well as the median and IQR of  $\mathcal{E}_{repr}$  with an optimal  $\sigma$  tuned for each method. Again, R-DPCP consistently yields the smallest reprojection error on all datasets. Moreover, R-DPCP seems to be less sensitive to the choice of threshold, as for example in Hpatches, the reprojection error only increases from 0.55px to 0.56px as  $\sigma$  goes from  $10^{-2.4}$  to  $10^{-1.6}$  ( $\epsilon$  being  $\sim 6\times$  larger), while that of R-DLT increases from 0.62px to 0.74px. Finally, R-DPCP has comparably small running time with other methods, e.g. below 5ms for most threshold choices.

#### 5.4 Homographic tensor estimation

**Data.** We use the TUM dataset [24], which consists of video sequences of indoor environments taken by a hand-held RGB-D camera. We take two sequences suitable for this task: ‘nostructure.texture\_near\_withloop’ (Near) where the scene is planar posters so that rich features can be detected and matched, and ‘360.hemisphere’ (Hemi) where the motion is almost pure rotation. We take a frame gap of 20 for ‘Near’ and 10 for ‘Hemi’ as used in [5]. A similar normalization is applied to correspondences in each view and the homographic tensor embeddings as in §5.1. As the geometrically related views increase from 2 to 3, the dimension of the optimization problem increases from 9 to 27. The task of homographic tensor estimation is more challenging than the case of two-views because a hyperplane in  $\mathbb{R}^{27}$  is less distinguishable than in  $\mathbb{R}^9$ .

**Methods.** The minimal sampler (Algorithm 1, line 5) takes 4 correspondences to linearly estimate a homographic tensor. Since each homographic tensor can be decomposed into two pairwise homography matrices, the reprojection error is used as a scoring function for RANSAC.

**Results.** We report the average angular rotation and angular translation errors ( $\mathcal{E}_{Rot}$ ,  $\mathcal{E}_{tran}$ ) in degrees with respect to different thresholds in Figure 9. As a first obser-

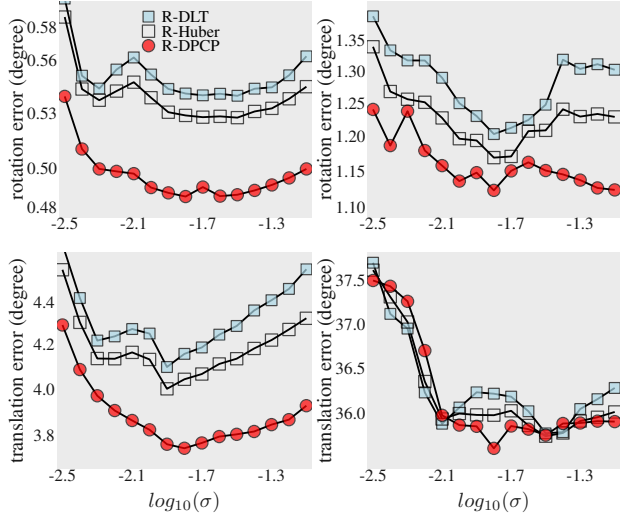


Figure 9: Homographic tensor estimation using 31 image triples from ‘Near’ (left) and 73 from ‘Hemi’ (right), with varying threshold multiplier  $\sigma$ . All methods terminate when a confidence of  $p = 95\%$  has been reached. Experiments are averaged over 100 trials. Rotation and translation errors are in degrees.

vation, in the downstream task of retrieving relative poses, R-DPCP produces the most accurate estimate. In ‘Near’, it yields 0.49° and 3.81° of rotation and translation error compared with 0.54° and 4.10° given by R-Huber. Interestingly, the translation errors given by all methods are significant ( $\geq 30^\circ$ ) on ‘Hemi’. This is because the inter-frame translation (baseline) for the sequence ‘Hemi’ is small. As such, the signal-to-noise ratio for translation is small, leading to a large translation error.

## 6 Conclusion

We employed the robust subspace learning method Dual Principal Component Pursuit inside the local optimization step of RANSAC. The proposed DPCP-RANSAC has far fewer parameters than existing methods [4, 11, 19, 1] and can be extended to various geometric vision tasks conveniently. Our experimental evaluation on large-scale datasets shows that DPCP-RANSAC (1) offers a stable estimation that is more accurate than its recent predecessors, (2) is scalable with time complexity comparable to that of its simplest alternatives, and (3) its sensitivity to the threshold parameter is significantly improved.

## References

- [1] Daniel Barath and Jiri Matas. Graph-Cut RANSAC. *IEEE Computer Society Conference on Computer Vision and Pat-*

- tern Recognition, pages 6733–6741, 2018. 1, 2, 3, 4, 5, 6, 8
- [2] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006. 5, 7
- [3] Ondřej Chum and Jiří Matas. Optimal randomized RANSAC. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(8):1472–1482, 2008. 4
- [4] Ondřej Chum, Jiří Matas, and Josef Kittler. Locally optimized RANSAC. In *German Conference on Pattern Recognition*, volume 2781, pages 236–243. Lecture Notes in Computer Science, 2003. 1, 2, 3, 4, 5, 8
- [5] Tianjiao Ding, Yunchen Yang, Zhihui Zhu, Daniel P. Robinson, René Vidal, Laurent Kneip, and Manolis C. Tsakiris. Robust Homography Estimation via Dual Principal Component Pursuit. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6079–6088, 2020. 1, 2, 3, 6, 8
- [6] Tianyu Ding, Zhihui Zhu, Tianjiao Ding, Yunchen Yang, René Vidal, Manolis C. Tsakiris, and Daniel P. Robinson. Noisy dual principal component pursuit. In *International Conference on Machine Learning*, pages 2910–2918, 2019. 1, 2, 3
- [7] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 1
- [8] R. Hartley and R. Vidal. The multibody trifocal tensor: motion segmentation from 3 perspective views. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, 2004. 3
- [9] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 5, 7
- [10] R. I. Hartley. In defence of the 8-point algorithm. In *Proceedings of IEEE International Conference on Computer Vision*, pages 1064–1070, 1995. 5
- [11] Karel Lebeda, Jiri Matas, and Ondrej Chum. Fixing the locally optimized RANSAC. *British Machine Vision Conference 2012*, pages 3–7, 2012. 1, 2, 3, 4, 5, 8
- [12] Karel Lebeda, Jiri Matas, and Ondrej Chum. Fixing the locally optimized ransac—full experimental evaluation. 2012. 4
- [13] Gilad Lerman and Tyler Maunu. An Overview of Robust Subspace Recovery. *Proceedings of the IEEE*, 106:1380–1410, 2018. 2
- [14] Xiao Li, Shixiang Chen, Zengde Deng, Qing Qu, Zhihui Zhu, and Anthony Man Cho So. Weakly Convex Optimization over Stiefel Manifold Using Riemannian Subgradient-Type Methods. pages 1–27, 2019. 2
- [15] Jiří Matas and Ondřej Chum. Randomized RANSAC with sequential probability ratio test. *Proceedings of the IEEE International Conference on Computer Vision*, II:1727–1732, 2005. 4
- [16] Tyler Maunu, Teng Zhang, and Gilad Lerman. A well-tempered landscape for non-convex robust subspace recovery. *Journal of Machine Learning Research*, 20:1–59, 2019. 2
- [17] Qing Qu, Ju Sun, and John Wright. Finding a sparse vector in a subspace: Linear sparsity using alternating directions. In *Advances in Neural Information Processing Systems 27*, pages 3401–3409. 2014. 2
- [18] Rahul Raguram, Jan Michael Frahm, and Marc Pollefeys. A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5303 LNCS(PART 2):500–513, 2008. 3
- [19] Jonas Serych, Jiri Matas, and Ondrej Drbohlav. Fast L1 - based RANSAC for homography estimation. *Computer Vision Winter Workshop*, 2016. 4, 8
- [20] Amnon Shashua and Michael Werman. Trilinearity of three perspective views and its associated tensor. In *IEEE International Conference on Computer Vision*, pages 920–925, 1995. 3
- [21] Amnon Shashua and Lior Wolf. Homography tensors: On algebraic entities that represent three views of static or moving planar points. In *European Conference on Computer Vision*, pages 507–521. Springer, 2000. 3
- [22] Henrik Stewenius, Christopher Engels, and David Nistér. Recent developments on direct relative orientation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60(4):284–294, 2006. 3, 4, 6
- [23] Christoph Strecha, Wolfgang Von Hansen, Luc Van Gool, Pascal Fua, and Ulrich Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. Ieee, 2008. 5
- [24] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *International Conference on Intelligent Robot Systems*, Oct. 2012. 8
- [25] Manolis C. Tsakiris and René Vidal. Dual principal component pursuit. In *ICCV Workshop on Robust Subspace Learning and Computer Vision*, pages 10–18, 2015. 1, 2, 3
- [26] Manolis C. Tsakiris and René Vidal. Hyperplane clustering via dual principal component pursuit. *International Conference on Machine Learning*, 7:5328–5337, 2017. 2
- [27] Manolis C. Tsakiris and René Vidal. Dual principal component pursuit. *Journal of Machine Learning Research*, 19:1–49, 2018. 1, 2, 3
- [28] René Vidal and John Oliensis. Structure from planar motions with small baselines. In *European conference on computer vision*, pages 383–398, 2002. 7
- [29] Zhihui Zhu, Tianyu Ding, Daniel Robinson, Manolis C. Tsakiris, and René Vidal. A Linearly Convergent Method for Non-Smooth Non-Convex Optimization on the Grassmannian with Applications to Robust Subspace and Dictionary Learning. In *Advances in Neural Information Processing Systems 32*, pages 9442–9452. 2019. 2
- [30] Zhihui Zhu, Yifan Wang, Daniel Robinson, Daniel Naiman, René Vidal, and Manolis C. Tsakiris. Dual principal component pursuit: Improved analysis and efficient algorithms. *Advances in Neural Information Processing Systems*, 2018-Decem(NeurIPS):2171–2181, 2018. 1, 2, 3