Space-Partitioning RANSAC

Daniel Barath[1] and Gabor Valasek[2]

[1] ETH Zürich, Computer Vision and Geometry Group, Switzerland
`danielbela.barath@inf.ethz.ch`
[2] Eötvös Loránd University, Budapest, Hungary
`valasek@inf.elte.hu`

Abstract. A new algorithm is proposed to accelerate the RANSAC model qual-ity calculations. The method is based on partitioning the joint correspondence space, e.g., 2D-2D point correspondences, into a pair of regular grids. The grid cells are mapped by minimal sample models, estimated within RANSAC, to re-ject correspondences that are inconsistent with the model parameters early. The proposed technique is general. It works with arbitrary transformations even if a point is mapped to a point set, e.g., as a fundamental matrix maps to epipo-lar lines. The method is tested on thousands of image pairs from publicly avail-able datasets on fundamental and essential matrix, homography and radially dis-torted homography estimation. On average, the proposed space partitioning algo-rithm reduces the RANSAC run-time by 41% with provably no deterioration in the accuracy. When combined with SPRT, the run-time drops to its 30%. It can be straightforwardly plugged into any state-of-the-art RANSAC framework. The code is available at https://github.com/danini/graph-cut-ransac.

Keywords: RANSAC, preemptive verification, space partitioning

1 Introduction

Robust model estimation is a fundamental problem in computer vision, allowing the es-timation of accurate geometric models, *e.g.* relative camera pose, from noisy input data. The RANdom SAmple Consensus (RANSAC) algorithm, proposed by Fischler and Bolles [19], has become one of the most prominent robust estimators with a number of real-world applications. For instance, it is used for building large-scale 3D recon-structions by running in state-of-the-art structure-from-motion [42,41] (SfM) or simul-taneous localization and mapping [33,10] (SLAM) algorithms. It is used for pose-graph initialization in global SfMs [7], for wide baseline image matching [37,31,32], multi-model fitting [56,23,35,3], and image mosaicing [21,17].

While being one of the most successful robust estimators in vision, the RANSAC algorithm iterates three conceptually simple steps. First, it randomly selects a minimal sample of data points, *e.g.*, five 2D-2D point correspondences. Second, it estimates the model parameters, *e.g.*, essential matrix, from the minimal sample. Third, it measures the model quality as the cardinality of its support, *i.e.*, the number of points closer than a manually set inlier-outlier threshold. Even though several improvements have been proposed throughout the years, the basic structure of the algorithm remained the same. In this paper, we focus on speeding up the model quality calculation via partitioning the

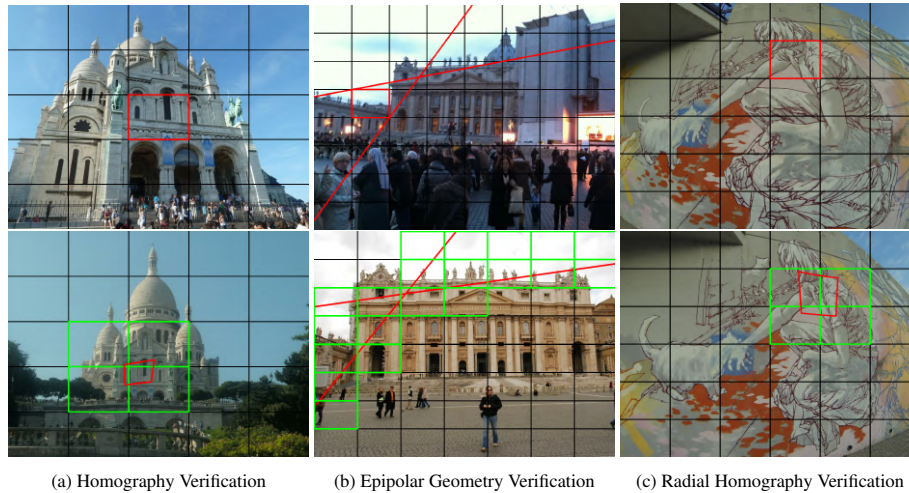| (a) Homography Verification | (b) Epipolar Geometry Verification | (c) Radial Homography Verification |

Fig. 1. Examples of the proposed space partitioning-based model quality calculation. For a cell (red rectangle) in the top image, only those correspondences are checked where the point is in the rectangle and its corresponding pair falls inside a green rectangle in the bottom image.

correspondences into pairs of n-dimensional cells, and we select the potential inliers extremely efficiently before computing the quality of each candidate model.

Many recent variants of RANSAC focus on improving the accuracy by locally optimizing each so-far-the-best model [16,28,2,5,4]. This local optimization can be, *e.g.*, an iteratively re-weighted least-squares procedure [5,4], or an inner RANSAC [16,28] that exploits additional global information, *e.g.*, spatial structures in the data [2]. Some algorithms aim at achieving higher accuracy by learning [6] or better modelling [47,46,5,4] the noise in the input data. Also, several methods have been proposed in the recent years including deep learning in the RANSAC procedure as an inlier probability predictor [52,39,8,54,43,55,44] that helps filtering outliers. As it is shown in [25], such methods are essential to have state-of-the-art accuracy on real world data.

There have also been a lot of papers published aiming to accelerate the robust estimation procedure. Most commonly, researchers focus on improving the sampling process to increase the probability of finding a good sample early and, thus, a model that triggers the termination criterion. The NAPSAC [45] sampler assumes that inliers are spatially coherent. The first point in the sample is selected randomly. The others are from a hyper-sphere centered on the first location-defining one. The GroupSAC algorithm [34] is built on the assumption that the inliers of a model have "similar" properties and, therefore, can be separated into groups prior to the estimation. The PROSAC [14] algorithm exploits a predicted inlier probability rank of each point. It starts the sampling from the most promising ones and then progressively increases the sampling pool. Progressive NAPSAC [4] combines the advantages of local and global sampling by drawing samples from a gradually growing hyper-sphere. Recently, papers also focus on rejecting likely ill-conditioned or degenerate minimal samples early [6,12] to avoid estimating the model parameter unnecessarily.

Another way of making the procedure more efficient is to avoid unnecessary calculations when computing the quality of a candidate model. In most robust estimators, the quality calculation is done for every estimated model by computing *all* point-to-model residuals. This procedure is generally of $\mathcal{O}(NK)$ complexity, where N is the number of input data points and K is the number of models generated inside RANSAC. In the case of having thousands of input data points or a low inlier ratio, the quality calculation dominates the run-time of the robust estimation. To tackle this issue, preemptive model verification strategies have been proposed to recognize incorrect models early during the model quality calculation. The $T_{D,D}$ test [13] runs the model verification on $D \in \mathbb{N}^+$ points selected at random (where $D \ll N$). The rest of the data points are checked only if the evaluated D points are all inliers. The method was extended by the bail-out test [11] that calculates the inlier ratio on a subset of the data points. If the ratio on the subset is significantly lower than the so-far-the-best one, the probability of the current model being better than the previous best is small and thus, the model is rejected early. In [30,15], Wald's Sequential Probability Ratio Test (SPRT) [50] is used to develop an optimal model verification method. The SPRT algorithm solves a constrained optimization problem, where the user supplies acceptable probabilities for errors in rejecting a good model and accepting a bad one. The resulting optimal test is a trade-off between the time spent before a decision and the errors committed.

These methods, however, do not exploit that in computer vision the estimated model is usually an $\mathbb{R}^n \to \mathbb{R}^m$ mapping defined on geometrically interpretable data points, such as 2D-2D point correspondences. This property allows us to partition the input points by bounding structures, *e.g.* n-dimensional axis-aligned boxes (AABB), and to define the sought model as a mapping between the bounded domains. This is an extremely efficient way of selecting candidate inliers without calculating the point-to-model residuals of the rejected points. The benefit is two-fold: first, significantly fewer points are needed to be tested when calculating the model quality. Second, it allows an early model rejection if the number of selected candidate inliers is lower than that of the so-far-the-best model. Moreover, the proposed algorithm is *general*. It works for all kinds of models used in computer vision, *e.g.*, rigid motion, homography, epipolar geometry. It can be straightforwardly included in state-of-the-art frameworks, *e.g.* VSAC [24], and be combined with its "bells and whistles", *e.g.*, the SPRT test [30,15].

2   Background and Problem Formulation

Recent robust model fitting algorithms [38,27,9,2,5,4,24] spend a considerable amount of time calculating the point-to-model residuals when selecting the inliers of each verified model. The objective of this paper is to speed up the quality metric calculation by conservatively filtering out correspondences that are guaranteed to be outliers.

For simplicity, we explain the idea through a simple example before formalizing it in a general way. Assume that we are given a homography, estimated from four point correspondences between two images, and the objective is to calculate the support of the model, *i.e.*, the number of inliers. As a preliminary step, we partition the points in each image to a regular grid, *i.e.*, we have two grids in total. To determine the number of inliers, we process each cell of the first grid. Each cell stores a set of $(\boldsymbol{p}, \boldsymbol{q})$ corre-
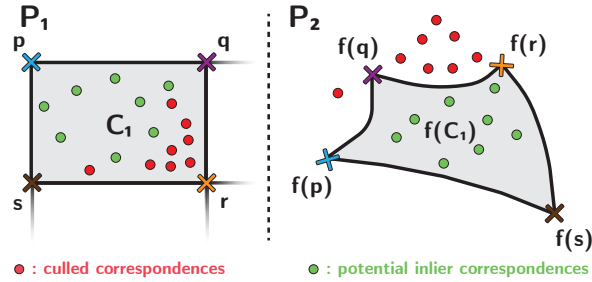
Fig. 2. Cell C_1 in image plane P_1 and its projection $f(C_1)$ by model $\boldsymbol{f}$ to P_2. A correspondence $(\boldsymbol{p}, \boldsymbol{q})$ with $\boldsymbol{q}$ in P_2 falling outside $\boldsymbol{f}(C_1)$ is rejected (red dots). The potential inliers (green dots) of $\boldsymbol{f}$ are the ones where $\boldsymbol{p}$ falls inside C_1 and $\boldsymbol{q}$ in its image $\boldsymbol{f}(C_1)$.

spondences between the two images. We keep all $(\boldsymbol{p}, \boldsymbol{q})$ pairs for which $\boldsymbol{q}$ is inside the projected image of the cell under the homography. More precisely, only those $(\boldsymbol{p}, \boldsymbol{q})$ correspondences are kept as candidate inliers, where $\boldsymbol{q}$ is in a cell in the second image which overlaps with the projection (by the homography) of the cell in which $\boldsymbol{p}$ falls. See Figures 1, 2 for examples.

Our formal description incorporates classes of models that map sets of points to points, so that we may address problem such as epipolar geometry estimation, where epipolar lines are mapped to points [22]. This comes at the expense of notational complexity but it also demonstrates the generality of our method.

General formulation. Let $\mathcal{S} = \{(\boldsymbol{p}_i, \boldsymbol{q}_i) \in \mathbb{R}^n \times \mathbb{R}^m \mid i = 1, .., N\}$ be a set of correspondences. Points from the domain, or first image, are denoted by $\boldsymbol{p}_i \in \mathbb{R}^n$ and points in the range, *i.e.* from the second image, are written as $\boldsymbol{q}_i \in \mathbb{R}^m$. Set $\mathcal{S}$ may consist of 2D-2D point pairs ($n = m = 2$) found in two images and then used for estimating a homography or the epipolar geometry [22], 2D-3D correspondences ($n = 2, m = 3$) used to solve the perspective-n-point problem [53], or 3D-3D ones ($n = m = 3$) for point cloud registration [36]. Let us assume that these correspondences stem from a $\boldsymbol{g} : \mathbb{R}^n \to \mathcal{P}(\mathbb{R}^m)$ ground truth mapping, where $\mathcal{P}(\mathbb{R}^m)$ denotes the power set of $\mathbb{R}^m$, *i.e.*, the set of all possible subsets of $\mathbb{R}^m$. Usually, the range sets consist of a single point or posses a simple structure. For example, they form lines in the second image, such as in the case of estimating epipolar geometry where a point is mapped to an epipolar line [22]. Due to various sources of error, *e.g.* measurement and quantization, only $\boldsymbol{q}_i \approx \boldsymbol{g}(\boldsymbol{p}_i)$ holds.

Our objective is to find an $\boldsymbol{f} : \mathbb{R}^n \to \mathcal{P}(\mathbb{R}^m)$ *model mapping* from the $(\boldsymbol{p}_i, \boldsymbol{q}_i)$ correspondences that best approximates $\boldsymbol{g}$ in the sense that it produces a maximal inlier set for a given $\epsilon > 0$ inlier threshold. That is, we seek the largest cardinality of the set

$$\mathcal{I}_{\boldsymbol{f}} = \{(\boldsymbol{p}_i, \boldsymbol{q}_i) \in \mathcal{S} \mid \exists \boldsymbol{q} \in \boldsymbol{f}(\boldsymbol{p}_i) : |\boldsymbol{q} - \boldsymbol{q}_i| < \epsilon\}, \tag{1}$$

where norm $|\cdot| : \mathbb{R}^m \to \mathbb{R}$ is some distance function defined on the points, *e.g.*, re-projection error for homographies. This is the robust model fitting problem. Note that while our model quality function maximizes the inlier number, it is straightforward to use our proposed method with state-of-the-art functionals as well, such as the truncated

---

Algorithm 1 Inlier prefiltering with conservative rejection

1: **Input**: model $\boldsymbol{f} : \mathbb{R}^n \to \mathcal{P}(\mathbb{R}^m)$; inlier threshold ϵ; $\mathcal{C}_1, \mathcal{C}_2$ partitionings in images P_1, P_2.
2: **Output**: potential inliers $\mathcal{I}$
3: $\mathcal{I} \leftarrow \varnothing$
4: **for** $\forall C_1 \in \mathcal{C}_1$ **do**   ▷ Iterate through the 2D cells in P_1. Section 3.1.
5:     $B \leftarrow \text{Bound}(\boldsymbol{f}(C_1), \epsilon)$ ▷ Bounding structure of $\boldsymbol{f}(C_1)$ in P_2. Section 3.2.
6:     **for** $\forall C_2 \in \mathcal{C}_2$ **do**
7:         **if** $C_2 \cap B \neq \varnothing$ **then**   ▷ Cells that intersect with B in P_2.
8:             **for** $\forall (\boldsymbol{p}_i, \boldsymbol{q}_i) \in (C_1, C_2)$ **do**   ▷ Correspondences falling inside cells (C_1, C_2).
9:                 $\mathcal{I} \leftarrow \mathcal{I} \cup \{(\boldsymbol{p}_i, \boldsymbol{q}_i)\}$

---

quadratic loss of MSAC [47] or that of MAGSAC [5] and MAGSAC++ [4] marginalizing over a range of noise scale.

3   Correspondence Culling

Let $\mathcal{P}_1 \subset \mathbb{R}^n$ denote the domain of the $\boldsymbol{f} : \mathbb{R}^n \to \mathcal{P}(\mathbb{R}^m)$ model mappings and $\mathcal{P}_2 \subset \mathbb{R}^m$ the range. Our proposed method partitions the correspondences based on a spatial subdivision of $\mathcal{P}_1$ into cells, for example into a regular grid. We show that the image of such regions may be conservatively bounded for a wide range of model mappings in the sense that it is guaranteed that no point of the cell may map outside of this bound. A $(\boldsymbol{p}_i, \boldsymbol{q}_i)$ correspondence is culled if $\boldsymbol{q}_i \in \mathcal{P}_2$ is outside of the bounded image of the cell containing $\boldsymbol{p}_i \in \mathcal{P}_1$. The resulting set of nD-mD point correspondences is further processed by computing the exact inlier count or model score.

Alg. 1 summarizes our approach. For the sake of simplicity, we will describe each step assuming that we are given 2D-2D point correspondences. Nevertheless, the algorithm is general and, thus, also works with other data types. In Step 4, we iterate through the cells in $\mathcal{C}_1$, *i.e.*, the partitioning in domain $\mathcal{P}_1$ (*i.e.*, the first image). Next, we calculate the bounding structure $B(\boldsymbol{f}(C_1), \epsilon)$ in domain $\mathcal{P}_2$ (*i.e.*, the second image) of the current cell C_1. In Steps 6 and 7, we iterate through all cells in $\mathcal{C}_2$ and select those which intersect with $B(\boldsymbol{f}(C_1), \epsilon)$. Assuming that $\mathcal{C}_2$ consists of axis-aligned boxes (*e.g.*, it is a uniform grid or a quad-tree) and $B(\boldsymbol{f}(C_1), \epsilon)$ is an axis-align bounding box, this step simply calculates the intersection of two axis-aligned rectangles that has negligible time demand. In Step 8, we iterate through all $(\boldsymbol{p}_i, \boldsymbol{q}_i)$ correspondences where $\boldsymbol{p}_i$ falls inside C_1 and $\boldsymbol{q}_i$ is in C_2. Note that this step does not require checking all correspondences. Point correspondence $(\boldsymbol{p}_i, \boldsymbol{q}_i)$ can be considered as a 4D point $\boldsymbol{s}_i$ in the concatenated space $\mathbb{R}^{n+m}$, where $n = m = 2$. Cell correspondence (C_1, C_2) is basically a 4D box C_{12} and, thus, the containment test degrades to checking if $\boldsymbol{s}_i$ falls inside C_{12}. When using hash maps and suitable hashing functions this step has $\mathcal{O}(1)$ complexity.

3.1   Correspondence Partitioning

Let us partition $\mathcal{P}_1$ into a disjoint set of $C_j \in \mathcal{C}$ cells ($j \in \mathbb{N}_{>0}$) and store the correspondences based on these, *i.e.* let $\mathcal{S}_j = \{(\boldsymbol{p}_i, \boldsymbol{q}_i) \in \mathcal{S} \mid \boldsymbol{p}_i \in C_j\}$. Set $\mathcal{S}_j$ consists

of the correspondences where the first point falls inside cell C_j in $\mathcal{P}_1$. In our particular case, we use a regular grid of equally sized axis-aligned rectangular cells for the subdivision. Partitioning is a pre-processing step with $\mathcal{O}(N)$ complexity and it only needs to be computed once, for example, upon reading the point correspondences. Decomposing $\mathcal{S}$ into the $\mathcal{S}_j$ partitions is not strictly necessary, however, it is an important component of efficient culling.

Note that the proposed method can be also used with more advanced space partitioning structures, such as quad-trees. We, however, empirically found that the required computational overhead is too large for the typical computer vision problems consisting of, at most, a few thousands of data points. This overhead stems from the tree construction and the increased number of cells to be projected by the proposed algorithm.

3.2   Culling by Images of Cells

Suppose that we are given a model $\boldsymbol{f} : \mathcal{P}_1 \to \mathcal{P}_2$ estimated, for example, from a minimal sample. The image of a $C_j \subset \mathcal{P}_1$ cell depends on the current model $\boldsymbol{f}$ and cannot be pre-computed. Consequently, we have to devise efficient means to compute or estimate this image, as it is calculated at every RANSAC iteration.

Depending on the algebraic properties of the model mapping, there are two cases to consider. If $\boldsymbol{f}$ is invertible, we cull $(\boldsymbol{p}_i, \boldsymbol{q}_i), \boldsymbol{p}_i \in C_j$ if $\boldsymbol{f}^{-1}$ does not map $\boldsymbol{q}_i$ into C_j, that is, if $\boldsymbol{f}^{-1}(\boldsymbol{q}_i) \notin C_j$. If $\boldsymbol{f}$ is not invertible, we bound the image of the C_j cell under $\boldsymbol{f}$ with some $B \subset \mathcal{P}_2$ and reject $(\boldsymbol{p}_i, \boldsymbol{q}_i)$ if $\boldsymbol{q}_i \notin B$ holds. This is further decomposed into two cases: B may be either an approximative or a conservative bound.

Invertible Case. Let us define the image of cell C_j as the image of all points in C_j, *i.e.*

$$\boldsymbol{f}(C_j) = \{\boldsymbol{y} \in \mathbb{R}^m \mid \exists \boldsymbol{x} \in C_j : \boldsymbol{y} \in \boldsymbol{f}(\boldsymbol{x})\}. \tag{2}$$

Note that $\boldsymbol{x}$ and $\boldsymbol{y}$ denote arbitrary points in $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively, and not necessarily the input correspondences.

Let us assume that there exists an $\boldsymbol{f}^{-1}$ inverse of the model map. The $\boldsymbol{f}^{-1}(\boldsymbol{q}_i) \in C_j$ containment is trivially resolved if we have a regular grid in $\mathcal{P}_1$. In the general case, let us assume that C_j is written as the intersection of a finite number of

$$Q_{kj} = \{\boldsymbol{x} \in \mathcal{P}_1 \mid q_{kj}(\boldsymbol{x}) \le 0\} \tag{3}$$

volumes, *i.e.* $C_j = \cap_k Q_{kj}$, where $q_{kj} : \mathcal{P}_1 \to \mathbb{R}$ is the implicit representation of the j-th boundary volume. This intersection contains all points inside and on the boundary of C_j. For example, a 2D rectangular cell may be written as the intersection of four half-planes of the form $a_k x + b_k y + c_k \le 0, k \in \{1, 2, 3, 4\}$. Recalling that the intersection of implicitly defined volumes may be written as a maximum operation [40], a $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{S}_j$ correspondence is culled, if and only if $\max\{q_{kj}(\boldsymbol{f}^{-1}(\boldsymbol{y})\}_k > 0$. Thus, the containment test in the transformed space $\mathcal{P}_2$ is reduced to a test in $\mathcal{P}_1$.

Although closed-form inversion is a strict restriction, all non-singular projective transformations possess this property. While no speed-up is achieved by transforming all points inside a cell in this way, the definitions will later be important to efficiently cull against AABBs of the cell boundaries.

General Case. If no closed-form inverse is available for $\boldsymbol{f}$, we may bound the image of cell C_j by using polynomial approximations of the transformed boundaries. We propose to use Lagrange interpolation at Chebyshev nodes to obtain these approximations. Once these polynomial approximations are obtained, we convert them to Bernstein basis and use the resulting Bézier control points to form an AABB bound of the cell image. This bounding characteristic follows from the fact that Bézier curves possess the convex hull property [18], that is, the entire image of the cell under the approximations should lie within the convex hull of the control points, and in turn, within their AABB.

In terms of specifics to achieve the above, let us assume that the boundaries of the transformed cell, $\boldsymbol{f}(C_j)$, are the images of the cell boundaries in C_j as $\boldsymbol{f}(\{\boldsymbol{x} \in \mathcal{P}_1 \mid q_{kj}(\boldsymbol{x}) = 0\})$. This merely simplifies the identification of the boundary curves of $\boldsymbol{f}(C_j)$. This is straightforward for models that map points to points as the polynomial approximations are traditional two or three dimensional Bézier curves. Models that map points to sets of points, *e.g.* epipolar geometry, can be embedded into this framework by assuming that there is a simple set of basis functions in which the image sets may be represented. For example, let us assume that the image sets are one parameter families and there exists a function basis that spans all sets. Formally, this means that, for all $\boldsymbol{x} \in \mathcal{P}_1$, we assume the existence of a $\boldsymbol{p}_{\boldsymbol{x}} : \mathbb{R} \to \mathcal{P}_2$ parametric mapping such that

$$\boldsymbol{f}(\boldsymbol{x}) = \{\boldsymbol{y} \in \mathcal{P}_2 \mid \exists t \in \mathbb{R} : \boldsymbol{y} = \boldsymbol{p}_{\boldsymbol{x}}(t)\} . \tag{4}$$

Let $\{e_i(t) : \mathbb{R} \to \mathbb{R}\}_{i=1}^k$ denote the basis that spans the images, *i.e.*, $\forall \boldsymbol{x} \in \mathcal{P}_1 : \forall i \in \{1, \dots, k\} : \exists \boldsymbol{a}_{\boldsymbol{x},i} \in \mathcal{P}_2 :$

$$\boldsymbol{p}_{\boldsymbol{x}}(t) = \sum_{i=1}^k a_{\boldsymbol{x},i} e_i(t) . \tag{5}$$

For example, lines in $\mathcal{P}_2$ may be represented in the $\{1, t\}$ basis as $\boldsymbol{p}_{\boldsymbol{x}}(t) = \boldsymbol{p}_{\boldsymbol{x},0} + t\boldsymbol{v}_{\boldsymbol{x}}$. The polynomial approximation of the boundary is reduced to the construction of a higher dimensional Bézier curve that maps from $\mathbb{R}$ to $\mathbb{R}^{mk}$, *i.e.* the Bézier curve is used to approximate the $a_{\boldsymbol{x},i}$ coefficients. As shown in the example of epipolar geometry estimation, this can be made simpler for a particular problem.

Conservative Bounds. The polynomial approximations shown above can be made conservative by applying the Lagrange interpolation error term to them.

If an arbitrary boundary curve of cell C_j is parametrized over an interval $[a, b]$ by some $\boldsymbol{q}_{kj} : [a, b] \to \mathcal{P}_1$ ($k \in \{1, 2, 3, 4\}$) mapping, then interpolating $\boldsymbol{f} \circ \boldsymbol{q}$ at k Chebyshev nodes $\boldsymbol{f}(\boldsymbol{q}_{kj}(\boldsymbol{x}_l))$, where

$$\boldsymbol{x}_l = \frac{a+b}{2} + \frac{b-a}{2} \cos \frac{2l-1}{2k}\pi, l = 1, \dots, k \tag{6}$$

yields a polynomial that is within

$$\frac{1}{2^{k-1}k!} \left(\frac{b-a}{2}\right)^k \cdot \max_{\xi \in [a,b]} \|(\boldsymbol{f} \circ \boldsymbol{q})^{(k+1)}(\xi)\|_\infty \tag{7}$$

of a $(k + 1)$ times continuously differentiable $\boldsymbol{f} \circ \boldsymbol{q}$ mapping [48]. Thus, offsetting the AABB of the Bézier control points by this value ensures that no points in C_j may

---

Algorithm 2 Bound($\boldsymbol{f}(C_j), \epsilon$) with AABB

    Input: current candidate mapping $\boldsymbol{f} : \mathbb{R}^2 \to \mathcal{P}(\mathbb{R}^2)$;
           approximation degree k; C_j cell; $\epsilon > 0$

1: $x_{min,j}, y_{min,j} \leftarrow +\infty$
2: $x_{max,j}, y_{max,j} \leftarrow -\infty$
3: **for** $\forall e$ edge of C_j **do**
4:      $\boldsymbol{a}, \boldsymbol{b} \leftarrow$ the endpoints of edge e
5:      $\{\boldsymbol{x}_l \leftarrow \boldsymbol{a} + t_l(\boldsymbol{b} - \boldsymbol{a})\}_{l=1}^k$; t_l are Chebyshev nodes
6:      $\boldsymbol{c}(t) \leftarrow$ polynomial interpolating $\{\boldsymbol{f}(\boldsymbol{x}_l)\}_{l=1}^k$
7:      $\boldsymbol{b}_0, \ldots, \boldsymbol{b}_{k+1} \leftarrow$ Bézier control points of $\boldsymbol{c}(t)$
8:      $x_{min,j} \leftarrow \min\left\{ x_{min,j}, \min_{i=0}^{k+1}\{[1,0] \cdot \boldsymbol{b}_i\} \right\}$
9:      $y_{min,j} \leftarrow \min\left\{ y_{min,j}, \min_{i=0}^{k+1}\{[0,1] \cdot \boldsymbol{b}_i\} \right\}$
10:     $x_{max,j} \leftarrow \max\left\{ x_{min,j}, \max_{i=0}^{k+1}\{[1,0] \cdot \boldsymbol{b}_i\} \right\}$
11:     $y_{max,j} \leftarrow \max\left\{ y_{min,j}, \max_{i=0}^{k+1}\{[0,1] \cdot \boldsymbol{b}_i\} \right\}$
12: $M \leftarrow$ bound on $\|\boldsymbol{f}^{(k+1)}(\boldsymbol{a} + t(\boldsymbol{b} - \boldsymbol{a}))\|_\infty, t \in [0,1]$
13: $C \leftarrow \max_{\boldsymbol{x}} \Pi_{j=0}^k |x - t_j| = \frac{1}{2^{2k-1}k!}$
14: $\delta \leftarrow \epsilon + \|\boldsymbol{b} - \boldsymbol{a}\|_\infty \cdot M + C$
15: **return** $\left(\begin{bmatrix} x_{min,j} - \delta \\ y_{min,j} - \delta \end{bmatrix}, \begin{bmatrix} x_{max,j} + \delta \\ y_{max,j} + \delta \end{bmatrix} \right)$

---

map outside the AABB bounding the image $\boldsymbol{f}(C_j)$ of cell C_j. Moreover, to account for the inlier-outlier threshold of the robust estimation procedure set, the AABB should additionally be offsetted by the threshold.

The simplest case is when the cells are axis-aligned rectangles and the $\boldsymbol{q}_{kj}$ boundaries are linear interpolations between the vertices of the edges. Algorithm 2 summarizes the construction of conservative bounds to the images in such a configuration. This has to be run once for each cell. For each edge of the cell, it requires k evaluations of $\boldsymbol{f}$. Steps 6 and 7 can be carried out simultaneously by a multiplication of a $(k-1) \times (k+1)$ matrix with a $(k+1) \times 2$ matrix. Step 12 depends on the candidate mapping family. In the case when no closed-form solutions are available for the bound, one has to estimate it by numerical means. The bound on the image can be made tighter by using the second grid on $\mathcal{P}_2$ and select the cells that intersect the convex hull of the control points.

Note that when using regular grids, the proposed algorithm can be significantly sped up by projecting the boundaries shared by multiple cells only once.

3.3 Early Model Rejection

The proposed approach provides all cell correspondences that might contain inliers of the currently estimated model. Besides being extremely useful for the verification, it also helps in rejecting models early without calculating their quality. The total number of data points stored in the selected cells is basically an upper bound on the inlier number. In the case this bound does not exceed the inlier number of the previous so-far-the-best model, the current model can be immediately rejected as it will not have more inliers than the best model so far. Therefore, a model θ is rejected if $\epsilon_r |\mathcal{I}^*| > |\mathcal{I}_\theta|$,

where $\mathcal{I}^*$ and $\mathcal{I}_\theta$ are, respectively, the inlier sets of the so-far-the-best and the currently tested models. Parameter $\epsilon_r = 1$ provably leads to no accuracy change. However, in practice, ϵ_r can be set marginally higher as we will show in the experiments.

Note that this approach works for all quality functions where the model quality is calculated from points closer than a manually set threshold, *e.g.*, as in RANSAC, MSAC or even in MAGSAC++ which uses a maximum threshold.

4   Model Estimation Problems

We show how the described general algorithm can be used in computer vision tasks.

4.1   Homography Estimation

Given homography $\mathbf{H} \in \mathbb{R}^{3\times3}$, the implied relationship of the points in the two images is written as $\alpha\mathbf{H}\boldsymbol{x}_1 = \boldsymbol{x}_2$, where α is a scaling that is inverse proportional to the homogeneous coordinate. The implied mapping is as

$$\boldsymbol{f}_{\mathrm{hom}}(\boldsymbol{x}) = \begin{bmatrix} f_u(\boldsymbol{x}) \\ f_v(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{h}_1^{\mathrm{T}}\boldsymbol{x}/\boldsymbol{h}_3^{\mathrm{T}}\boldsymbol{x} \\ \boldsymbol{h}_2^{\mathrm{T}}\boldsymbol{x}/\boldsymbol{h}_3^{\mathrm{T}}\boldsymbol{x} \end{bmatrix},$$

where $\boldsymbol{h}_1, \boldsymbol{h}_2, \boldsymbol{h}_3 \in \mathbb{R}^3$ are the rows of $\mathbf{H}$, and $f_u(\boldsymbol{x})$, $f_v(\boldsymbol{x})$ are coordinate functions. Since $\mathbf{H}$ maps lines to lines and we use a regular grid, the culling algorithm can be simplified to using the AABB of the cell corners projected by $\boldsymbol{f}_{\mathrm{hom}}$. This is a degree-one polynomial approximation that is also exact.

4.2   Epipolar Geometry Estimation

Given fundamental matrix $\mathbf{F}$, the implied relationship of the points in the two images is written as $\boldsymbol{x}_2^{\mathrm{T}}\mathbf{F}\boldsymbol{x}_1 = 0$, meaning that point $\boldsymbol{x}_2$ in the second image must fall on the corresponding epipolar line $\mathbf{l}_2 = \begin{bmatrix} a_2 & b_2 & c_2 \end{bmatrix}^{\mathrm{T}} = \mathbf{F}\boldsymbol{x}_1$. To bound the problem, we can assume that our model maps to the angle of the epipolar line as follows:

$$\boldsymbol{f}_{\mathrm{epi}}(\boldsymbol{x}) = \tan^{-1}\frac{b_2(\boldsymbol{x})}{a_2(\boldsymbol{x})}, \quad \boldsymbol{f}_{\mathrm{epi}}^{-1}(\boldsymbol{x}) = \tan^{-1}\frac{b_1(\boldsymbol{x})}{a_1(\boldsymbol{x})},$$

where $\mathbf{l}_i(\boldsymbol{x}) = \begin{bmatrix} a_i(\boldsymbol{x}) & b_i(\boldsymbol{x}) & c_i(\boldsymbol{x}) \end{bmatrix}$ is the epipolar line implied by $\boldsymbol{x}$ in the ith image. Given cell C_j containing the epipolar angles, it can be straightforwardly seen that

$$\boldsymbol{f}_{\mathrm{epi}}(C_j) : [\min\{\alpha \in C_j\}, \max\{\alpha \in C_j\}] \to [\min\{\alpha \in \boldsymbol{f}_{\mathrm{epi}}(C_j)\}, \max\{\alpha \in \boldsymbol{f}_{\mathrm{epi}}(C_j)\}].$$

Due to the convexity of the problem and the used regular grid, it can be decomposed into two sub-problems. First, the $\min$ and $\max$ operations are performed only on the intersections of the boundaries, *i.e.*, the cell corners. Therefore, a cell is not culled if at least one epipolar angle implied by its corners fall inside interval $[\min\{\alpha \in \boldsymbol{f}_{\mathrm{epi}}(C_j)\}, \max\{\alpha \in \boldsymbol{f}_{\mathrm{epi}}(C_j)\}]$. Second, a cell is not culled if one of its boundary lines intersects with the epipolar lines implied by angles $\min\{\alpha \in \boldsymbol{f}_{\mathrm{epi}}(C_j)\}$ and $\max\{\alpha \in \boldsymbol{f}_{\mathrm{epi}}(C_j)\}$. This is a more efficient formulation of the problem than considering epipolar lines as point sets.

4.3   Radial Homography Estimation

We use the one-parameter division model [20], for modeling radial distortion, is of form $\boldsymbol{g}(\boldsymbol{x}, \lambda) = [u, v, 1 + \lambda(x^2 + y^2)]^{\mathrm{T}}$, where $\boldsymbol{x} = [u, v]^{\mathrm{T}}$ is an image point. Given homography $\mathbf{H} = [\boldsymbol{h}_1, \boldsymbol{h}_2, \boldsymbol{h}_3]^{\mathrm{T}}$ with rows $\boldsymbol{h}_1, \boldsymbol{h}_2, \boldsymbol{h}_3 \in \mathbb{R}^3$, the points in the images are related as $\lambda_2 \boldsymbol{g}(\boldsymbol{x}_2, \lambda_2) = \mathbf{H} \boldsymbol{g}(\boldsymbol{x}_1, \lambda_1)$, where $\boldsymbol{x}_i$ and λ_i are, respectively, the point and the distortion parameter in the ith image, $i \in [1, 2]$. This implies the following mapping function $\boldsymbol{f}_{\mathrm{rad}}(\boldsymbol{x}) = [f_u(\boldsymbol{x}), f_v(\boldsymbol{x})]^{\mathrm{T}}$, where $f_u, f_v : \mathbb{R} \to \mathbb{R}$ are coordinate functions as follows:

$$f_u(\boldsymbol{x}) = \frac{\boldsymbol{h}_1^{\mathrm{T}} \boldsymbol{g}(\boldsymbol{x})}{\boldsymbol{h}_3^{\mathrm{T}} \boldsymbol{g}(\boldsymbol{x})}, \quad f_v(\boldsymbol{x}) = \frac{\boldsymbol{h}_2^{\mathrm{T}} \boldsymbol{g}(\boldsymbol{x})}{\boldsymbol{h}_3^{\mathrm{T}} \boldsymbol{g}(\boldsymbol{x})}. \tag{8}$$

The conservative bounds described in the previous sections are calculated from (8).

5   Experiments

We compare the proposed space partitioning-based model verification technique to the traditional one and to the SPRT [15] algorithm, *i.e.*, the state-of-the-art preemptive model verification technique. Besides comparing to SPRT, we also show that combining the two methods is highly beneficial. For robust estimation, we use LO-RANSAC [16] with PROSAC [14] sampling. We do not report model accuracy since the proposed technique leads to exactly the same number of inliers as verifying all points.

Homography Estimation. For testing the methods on homography estimation, we used the datasets from CVPR tutorial *RANSAC in 2020* [1]. We used the inlier-outlier threshold for RANSAC tuned in [1]. We ran the method only on scene Sacre Coeur consisting of 4950 image pairs. To form tentative correspondences, we detect 8000 SIFT keypoints in both images, and use mutual nearest neighbor check with SNN ratio test [29] as suggested in [49]. The average inlier ratio of the tested dataset is 14% for **E/F** and 6% for **H** estimation ranging from 0.9% to 68%.

In Fig. 3c, the relative run-time is plotted as the function of the $\log_{10}$ iteration number that was used as a fixed iteration number for RANSAC. Early rejection was turned off. Each curve shows the results of using a regular grid with different number of cells shown in brackets. The proposed approach leads to a speed-up with all tested cell numbers. In this case, 256 cells (each image is divided into 4×4 cells) lead to the fastest procedure. The run-time drops to its 50% when doing $10\,000$ iterations.

Fig. 5a shows the cumulative distribution functions (CDF) of the run-times (in ms) of SPRT, the proposed and traditional algorithms, and the proposed method with SPRT. For these experiments, we ran RANSAC with its confidence set to 0.99 and max. iteration number to 5000. This max. iteration number is a strict upper bound, preventing RANSAC to run longer. We set early rejection threshold ϵ_r to 1.6. The proposed technique with SPRT runs, on avg. for 31.2 ms, while the avg. time of SPRT is 39.3 ms.

Fundamental Matrix Estimation. Same data is used as before. In Fig. 3a, the relative run-time (*i.e.*, ratio of the time of the proposed and traditional approaches), the number of points verified in total and the processing time (in seconds) are plotted as the function of the $\log_{10}$ iteration number – used as a fixed iteration number for RANSAC. The

(a) Fundamental matrix estimation   (b) Essential matrix estimation   (c) Homography estimation
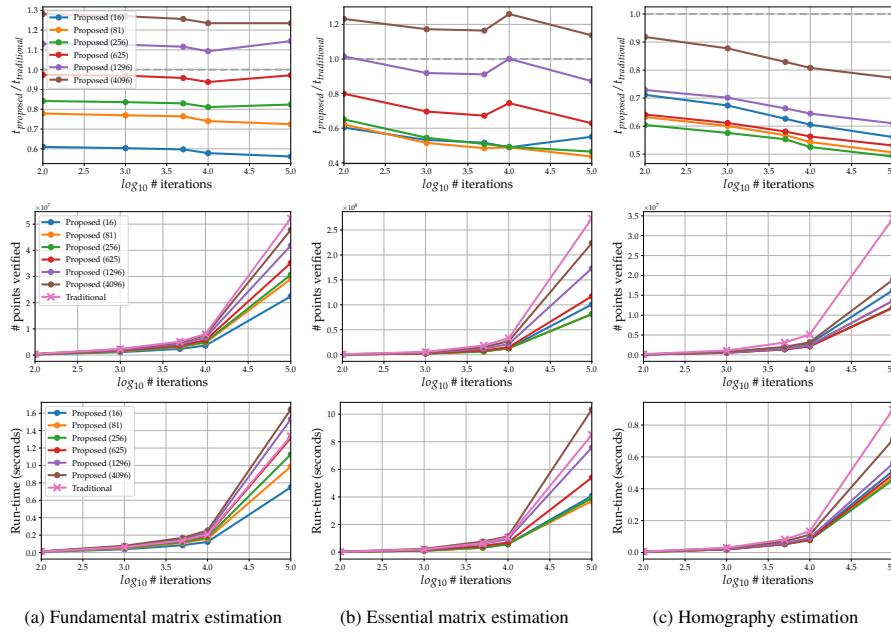
Fig. 3. The ratio of times of the proposed and traditional methods (top row; vertical axis), the number of points verified within the RANSAC loop (middle) and the run-time in seconds (bottom) are plotted as a function of $\log_{10}$ iteration number (horizontal). The number of 2D-2D cell pairs used in the proposed algorithm is written in brackets. For example, if the cell number is $2^4 = 16$, both images were divided into 2 pieces along each axis. We tested 2^4, 3^4, 4^4, 5^4, 6^4, and 8^4 subdivisions running the algorithm on a wide range of cell sizes.

proposed early rejection was turned off. Each curve shows the results of using a regular grid with different number of cells shown in brackets. For example, $16(= 2^4)$ means that each image axis is divided into 2 parts, thus, having 16 2D-2D cells in total. For **F** estimation, 16 cells lead to the fastest calculation with almost halving the run-time of the traditional approach. The time increases proportionally with the cell number. This is due to the fact that while having more cells provides a tighter approximation of the inlier set, it requires bounding more cells increasing the problem complexity. It is important to note that, similarly as for homography estimation, doing more RANSAC iterations and, thus, likely increasing the accuracy becomes cheaper with the proposed method.

Fig. 5b shows the cumulative distribution functions (CDF) of the processing times (in milliseconds) of SPRT, the proposed and traditional algorithms, and the proposed method with SPRT. We set the early rejection threshold ϵ_r to 1.2. The proposed approach with SPRT is the fastest by halving the run-time of the traditional approach and being, on average, faster by 50 ms than SPRT.

Essential Matrix Estimation. For essential matrix estimation, we used the same data as for fundamental matrices. In Fig. 3b, the relative run-time, the number of verified points and the actual run-time are plotted as the function of the $\log_{10}$ iteration number that was used as a fixed iteration number for RANSAC. Early rejection was turned off. Each
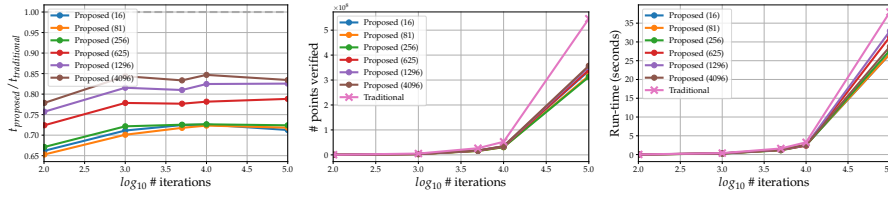
Fig. 4. The ratio of the run-times of the proposed and traditional methods, the number of points verified (middle) and the actual run-time (bottom) are plotted as a function of $\log_{10}$ iteration number for radial homography estimation on 16k image pairs from the Sun360 dataset.

curve shows the results of using a regular grid with different number of cells. Similarly as for fundamental matrix estimation, 16 cells lead to the fastest quality calculation. The speed-up is now even bigger: the run-time drops to its 40% when doing $10\,000$ iterations. This is caused by the fact that the five-point solver returns a maximum of 10 candidate solutions – the same number of RANSAC iterations requires more models to be verified than for fundamental matrix estimation.

Fig. 5c shows the cumulative distribution functions (CDF) of the processing times (in milliseconds) of SPRT, the proposed and traditional algorithms, and the proposed method with SPRT. We set early rejection threshold ϵ_r to 1.2. It can be seen that the proposed approach causes a quite significant speed-up. The proposed technique with SPRT runs, on average, for 317.1 ms, while the average time of SPRT is 416.2 ms.

Radial Homography Estimation. To test the proposed techniques on real-world data, we chose the Sun360 [51] panorama dataset. The purpose of the Sun360 database is to provide academic researchers a comprehensive collection of annotated panoramas covering 360×180-degree full view for a large variety of environmental scenes, places and the objects within. To build the core of the dataset, high-resolution panorama images were downloaded and grouped into different place categories. To obtain radially distorted image pairs from each $360°$ panoramic scene, we cut out images simulating a $80°$ FOV camera with a step size of $10°$ as done in [17]. Thus, the rotation around the vertical axis between two consecutive images is always $10°$. Finally, image pairs were formed by pairing the consecutive images in each scene. In total, 16056 image pairs were generated. For estimating radial distortion homographies from minimal samples, we use the solvers from [26]. See Fig. 6 for an example image stitching results using a radial homography on an image pair from the Sun360 dataset.

The effect of the grid density is shown in Fig. 4. The proposed approach accelerates the robust radial homography estimation on the tested wide range of cell numbers. The best run-times are achieved by partitioning the images into 3 pieces along each axis and, thus, having 81 cell correspondences in total.

Fig. 5d shows the cumulative distribution functions (CDF) of the processing times (in milliseconds) of SPRT, the proposed and traditional algorithms, and the proposed method with SPRT. We set early rejection threshold ϵ_r to 1.2. The proposed technique with SPRT runs, on average, for 134.7 ms, while the average time of SPRT is 331.3 ms which is almost three times higher than when using the proposed space partitioning.

(a) Homography Verification

(b) Fundamental Matrix Verification

(c) Essential Matrix Verification
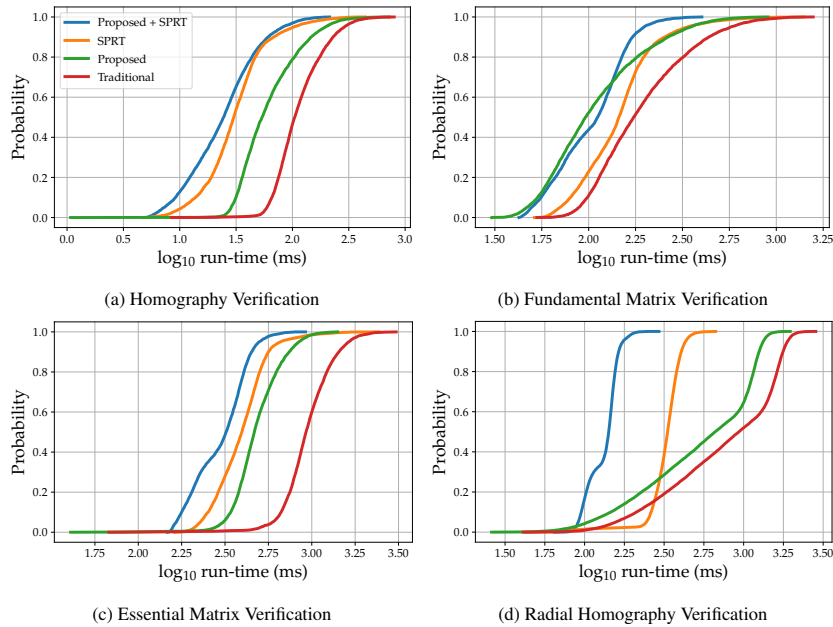
(d) Radial Homography Verification

Fig. 5. Cumulative distribution functions (CDF) of the times (ms) of the proposed and traditional algorithms, SPRT [15] and that of SPRT combined with the proposed method on homography, fundamental and essential matrix estimation on 4950 image pairs, and radial homography estimation on 16056 pairs. Being fast is indicated by a curve close to the top-left corner.

Timing Breakdown. We show the times spent on each steps of the robust estimation with and without SPRT when using the proposed space partitioning-based verification. The times and, also, the accuracy are shown in Table 1 on homography, essential and fundamental matrix estimation. The same datasets are used as in the previous sections. The cell rejection t_r has negligible time demand compared to the verification t_v. The verification time, when using the proposed approach, is significantly reduced. The AUC@10 scores are the same without SPRT and similar with SPRT.

Early Rejection. In the left plot of Fig. 6, the change in the run-time and the final inlier number is plotted as the function of the early rejection threshold ϵ_r. The results are divided by the result of the $\epsilon_r = 1$ case that provably does not lead to deterioration in the accuracy. The vertical lines are placed so the rejection threshold leads to lower than 1% drop in the inlier number. The green, orange and red lines overlap. For homographies, $\epsilon_r = 1.6$ leads to negligible accuracy drop while further decreasing the run-time by approximately 20%. For all other tested problems, setting ϵ_r to 1.2 is a reasonable choice decreasing the processing time by 22% and 9%, respectively.

6   Conclusion

We propose a new general algorithm for accelerating the RANSAC model quality calculation. The method is based on partitioning the joint correspondence space to a pair

Problem	SPRT	t_r	t_v	t_v^{trad}	AUC@10	AUC@10^{trad}
H	no	0.6	40.0	88.8	0.54	0.54
	yes	0.4	3.8	6.8	0.53	0.53
F	no	0.9	50.3	113.7	0.39	0.39
	yes	1.1	22.5	53.2	0.38	0.37
E	no	5.4	251.2	537.6	0.65	0.65
	yes	4.8	29.4	79.7	0.63	0.61

Table 1. The avg. (over all image pairs) time spent on cell rejection (t_r), model verification using the kept cells (t_v), in the traditional verification (t_v^{trad}) in *ms*; and the AUC@10 score of the max. of the rotation and translation errors, decomposed from homographies (**H**) fundamental (**F**) and essential matrices (**E**), when using the proposed and traditional approaches.
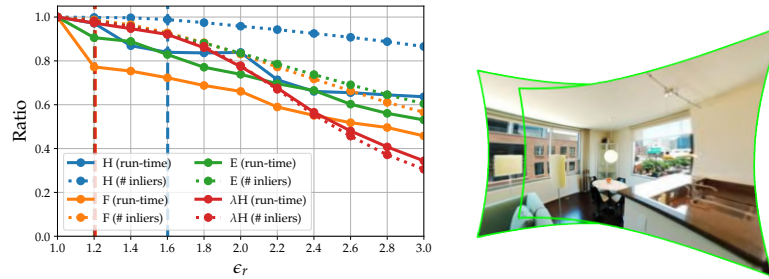

Fig. 6. **(Left)** Cumulative distribution functions (CDF) of the times (in ms) of the proposed and traditional techniques on homography (H), fundamental (F) and essential matrix (E) estimation on 4950 image pairs, and radial homography estimation (λH) on 16056 pairs. Being fast is indicated by a curve close to the top-left corner. **(Right)** Radial homography in the Sun360 dataset [51].

of regular grids. Cells of the grids are then projected by each minimal sample model, before calculating its quality, to efficiently reject all correspondences that are inconsistent with the model. Besides speeding up the quality calculation significantly, it also allows us to reject models early if the upper bound of their inlier number does not exceed the inlier number of the so-far-the-best model. We found that dividing the domain, *e.g.* images, into only a few cells is a good trade-off between getting a tight-enough approximation of the inlier set without significantly increasing the problem complexity.

The proposed technique reduces the RANSAC run-time by 41% on average on a wide range of problems and datasets. When it is combined with the SPRT test, it leads to an approximately 3.4 times speed-up compared to the traditional algorithm and, also, reduces the SPRT time to its 66%. It can be straightforwardly inserted into any state-of-the-art robust estimator, *e.g.*, VSAC [24] or MAGSAC++ [4], to accelerate them, provably, without any negative side-effect.

References

1. Barath, D., Chin, T.J., Chum, O., Mishkin, D., Ranftl, R., Matas, J.: RANSAC in 2020 tutorial. In: CVPR (2020), https://github.com/ducha-aiki/ransac-tutorial-2020-data 10
2. Barath, D., Matas, J.: Graph-cut RANSAC. In: CVPR. pp. 6733–6741 (2018) 2, 3
3. Barath, D., Matas, J.: Progressive-X: Efficient, anytime, multi-model fitting algorithm. In: ICCV. pp. 3780–3788 (2019) 1
4. Barath, D., Noskova, J., Ivashechkin, M., Matas, J.: MAGSAC++, a fast, reliable and accurate robust estimator. In: CVPR (2020) 2, 3, 5, 14
5. Barath, D., Noskova, J., Matas, J.: MAGSAC: marginalizing sample consensus. In: CVPR (2019), https://github.com/danini/magsac 2, 3, 5
6. Barath, D., Cavalli, L., Pollefeys, M.: Learning to find good models in RANSAC. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15744–15753 (2022) 2
7. Barath, D., Mishkin, D., Eichhardt, I., Shipachev, I., Matas, J.: Efficient initial pose-graph generation for global sfm. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14546–14555 (2021) 1
8. Brachmann, E., Rother, C.: Neural- Guided RANSAC: Learning where to sample model hypotheses. In: ICCV (2019) 2
9. Cai, Z., Chin, T.J., Koltun, V.: Consensus maximization tree search revisited. In: ICCV. pp. 1637–1645 (2019) 3
10. Campos, C., Elvira, R., Rodríguez, J.J.G., Montiel, J.M., Tardós, J.D.: Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. IEEE Transactions on Robotics 37(6), 1874–1890 (2021) 1
11. Capel, D.P.: An effective bail-out test for RANSAC consensus scoring. In: BMVC (2005) 3
12. Cavalli, L., Pollefeys, M., Barath, D.: NeFSAC: Neurally filtered minimal samples. In: The European Conference on Computer Vision (2022) 2
13. Chum, O., Matas, J.: Randomized RANSAC with tdd test. In: BMVC. vol. 2, pp. 448–457 (2002) 3
14. Chum, O., Matas, J.: Matching with PROSAC-progressive sample consensus. In: CVPR. IEEE (2005) 2, 10
15. Chum, O., Matas, J.: Optimal randomized RANSAC. TPAMI 30(8), 1472–1482 (2008) 3, 10, 13
16. Chum, O., Matas, J., Kittler, J.: Locally optimized ransac. In: Joint Pattern Recognition Symposium. Springer (2003) 2, 10
17. Ding, Y., Barath, D., Kukelova, Z.: Minimal solutions for panoramic stitching given gravity prior. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5579–5588 (October 2021) 1, 12
18. Farin, G.: Curves and Surfaces for Computer Aided Geometric Design (3rd Ed.): A Practical Guide. Academic Press Professional, Inc., San Diego, CA, USA (1993) 7
19. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM (1981) 1
20. Fitzgibbon, A.W.: Simultaneous linear estimation of multiple view geometry and lens distortion. In: CVPR. vol. 1, pp. I–I. IEEE (2001) 10
21. Ghosh, D., Kaabouch, N.: A survey on image mosaicking techniques. Journal of Visual Communication and Image Representation (2016) 1
22. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge university press (2003) 4

23. Isack, H., Boykov, Y.: Energy-based geometric multi-model fitting. IJCV (2012) 1
24. Ivashechkin, M., Barath, D., Matas, J.: VSAC: Efficient and accurate estimator for h and f (2021) 3, 14
25. Jin, Y., Mishkin, D., Mishchuk, A., Matas, J., Fua, P., Yi, K.M., Trulls, E.: Image matching across wide baselines: From paper to practice. IJCV **129**(2), 517–547 (2021) 2
26. Kukelova, Z., Heller, J., Bujnak, M., Pajdla, T.: Radial distortion homography. In: CVPR. pp. 639–647 (2015) 12
27. Le, H.M., Chin, T.J., Eriksson, A., Do, T.T., Suter, D.: Deterministic approximate methods for maximum consensus robust fitting. TPAMI (2019) 3
28. Lebeda, K., Matas, J., Chum, O.: Fixing the locally optimized RANSAC. In: BMVC. Citeseer (2012) 2
29. Lowe, D.G.: Object recognition from local scale-invariant features. In: ICCV. IEEE (1999) 10
30. Matas, J., Chum, O.: Randomized RANSAC with sequential probability ratio test. In: ICCV. vol. 2, pp. 1727–1732. IEEE (2005) 3
31. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. IVC (2004) 1
32. Mishkin, D., Matas, J., Perdoch, M.: MODS: Fast and robust method for two-view matching. CVIU (2015) 1
33. Mur-Artal, R., Tardós, J.D.: Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. IEEE transactions on robotics **33**(5), 1255–1262 (2017) 1
34. Ni, K., Jin, H., Dellaert, F.: GroupSAC: Efficient consensus in the presence of groupings. In: ICCV. pp. 2193–2200. IEEE (2009) 2
35. Pham, T.T., Chin, T.J., Schindler, K., Suter, D.: Interacting geometric priors for robust multimodel fitting. TIP (2014) 1
36. Pomerleau, F., Colas, F., Siegwart, R.: A review of point cloud registration algorithms for mobile robotics. Foundations and Trends in Robotics **4**(1), 1–104 (2015) 4
37. Pritchett, P., Zisserman, A.: Wide baseline stereo matching. In: ICCV. IEEE (1998) 1
38. Raguram, R., Chum, O., Pollefeys, M., Matas, J., Frahm, J.M.: USAC: a universal framework for random sample consensus. TPAMI (2013), `https://www.cs.unc.edu/~rraguram/usac` 3
39. Ranftl, R., Koltun, V.: Deep fundamental matrix estimation. In: The European Conference on Computer Vision (2018) 2
40. Ricci, A.: A Constructive Geometry for Computer Graphics. The Computer Journal **16**(2), 157–160 (May 1973). https://doi.org/10.1093/comjnl/16.2.157 6
41. Schönberger, J., Zheng, E., Pollefeys, M., Frahm, J.M.: Pixelwise view selection for unstructured multi-view stereo. In: ECCV (2016) 1
42. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: CVPR (2016) 1
43. Sun, W., Jiang, W., Tagliasacchi, A., Trulls, E., Yi, K.M.: Attentive context normalization for robust permutation-equivariant learning. In: CVPR (2020) 2
44. Tong, W., Matas, J., Barath, D.: Deep magsac++. arXiv preprint arXiv:2111.14093 (2021) 2
45. Torr, P.H., Nasuto, S.J., Bishop, J.M.: NAPSAC: High noise, high dimensional robust estimation-it's in the bag (2002) 2
46. Torr, P.H.S.: Bayesian model estimation and selection for epipolar geometry and generic manifold fitting. IJCV (2002) 2
47. Torr, P.H.S., Zisserman, A.: MLESAC: A new robust estimator with application to estimating image geometry. CVIU (2000) 2, 5
48. Trefethen, L.N.: Approximation Theory and Approximation Practice. SIAM (2012) 7
49. Trulls, E., Jun, Y., Yi, K., Mishkin, D., Matas, J., Fua, P.: Image matching challenge. In: CVPR (2020), `http://cmp.felk.cvut.cz/cvpr2020-ransac-tutorial/` 10

50. Wald, A.: Sequential analysis. Courier Corporation (2004) 3
51. Xiao, J., Ehinger, K.A., Oliva, A., Torralba, A.: Recognizing scene viewpoint using panoramic place representation. In: CVPR (2012), http://3dvision.princeton.edu/projects/2012/SUN360 12, 14
52. Yi*, K.M., Trulls*, E., Ono, Y., Lepetit, V., Salzmann, M., Fua, P.: Learning to find good correspondences. In: CVPR (2018) 2
53. Yuan, J.S.C.: A general photogrammetric method for determining object position and orientation. Transactions on Robotics and Automation **5**(2), 129–142 (1989) 4
54. Zhang, J., Sun, D., Luo, Z., Yao, A., Zhou, L., Shen, T., Chen, Y., Quan, L., Liao, H.: Learning two-view correspondences and geometry using order-aware network. ICCV (2019) 2
55. Zhao, C., Ge, Y., Zhu, F., Zhao, R., Li, H., Salzmann, M.: Progressive correspondence pruning by consensus learning. In: ICCV (2021) 2
56. Zuliani, M., Kenney, C.S., Manjunath, B.S.: The multiRANSAC algorithm and its application to detect planar homographies. In: ICIP. IEEE (2005) 1