

# ANTSAC: A Generic RANSAC Variant using Principles of Ant Colony Algorithms

Sebastian Otte\*, Ulrich Schwanecke† and Andreas Zell\*

\*Cognitive Systems Group  
Department of Compute Science  
University of Tuebingen  
Tuebingen, Germany

Email: firstname.lastname@uni-tuebingen.de

†Computer Vision and Mixed Reality Lab  
Department of Computer Science  
University of Applied Sciences Wiesbaden  
Wiesbaden, Germany

Email: ulrich.schwanecke@hs-rm.de

**Abstract**—In this paper, we present a new variant of the well-known RANdom SAMple Consensus (RANSAC) algorithm for robust estimation of model parameters. The idea of our method is based on a kind of volatile memory which is similar to the pheromone evaporation in the ant colony optimization algorithm. Therefore, we call our improved RANSAC like algorithm ANTSAC. We describe our new approach and the influence of its relevant parameters to the achieved performance in detail. ANTSAC is computationally efficient and convincingly easy to implement. It turns out that ANTSAC significantly outperforms RANSAC regarding the number of inliers after a given number of iterations. Further, we show that the advantage of ANTSAC increases with the complexity of the problem, i.e., with the number of model parameters, as well as with the relative number of outliers. ANTSAC is entirely generic, such that no further domain knowledge is required, as it is for many other RANSAC extensions. Nevertheless, we show that it is competitive to state-of-the-art methods even in domain specific scenarios.

## I. INTRODUCTION

Fitting a mathematical model to measured data is at the base of many applications in automated image analysis and robotics. In practice, the measurements are usually inaccurate and even erroneous. In computer vision tasks, for instance, data often come from error-prone feature detectors. Therefore algorithms fitting a model based on measured data must consider measuring errors. To be more precisely, such algorithms should separate accurate data (*inliers*) from less accurate data (*outliers*).

A common method for robust calculation of the model parameters is the non-deterministic iterative RANdom Sample Consensus (RANSAC) algorithm [1]. In each iteration of this algorithm a minimum set of points is selected, which is needed to fit the model. Based on this selection the model parameters are determined and subsequently all points are classified as inliers or outliers according to their distance to the estimated model. After a predefined number of iterations the model, that returned the highest number of inliers, is selected.

One major problem of RANSAC is that the number of iterations required to obtain an appropriate model explodes if the relative number of outliers or the number of model parameters increases. This fact is crucial for real-time applications, even more if instantiating the model is computationally expensive. If there is further knowledge of the model instance e.g. size, orientation etc., samples might be identified as outlier a priori. A couple of RANSAC modifications aim at gaining knowledge

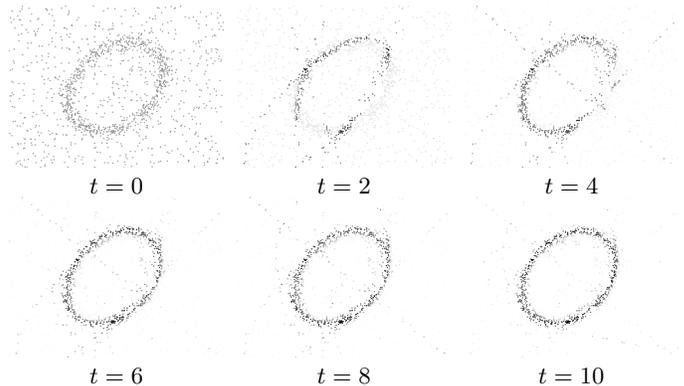


Fig. 1. Inlier emergence in a typical ellipse experiment during 10 iterations of ANTSAC. The intensity (darkness) of a point represents its pheromone-level.

by analyzing the sample set in a problem specific way or through density estimations. But such approaches are not always possible or even useful, especially if their advantages do not legitimate their computational overhead, not to mention the additional effort to implement and integrate them.

In this paper we present a generic improved RANSAC like algorithm that is based on collecting knowledge about the model instance during the iteration process itself, by memorizing, which samples have been proven to produce more inliers without any further data analysis. In order to achieve this we use some kind of volatile memory, which works similar to the pheromone evaporation process in the ant colony optimization algorithm [2]. We therefore call our approach ANTSAC. Figure 1 illustrates exemplary the emergence of an ellipse in the pheromone memory during 10 iterations of ANTSAC.

Based on experiments of notable problems in computer vision tasks we show that ANTSAC dramatically outperforms RANSAC both in terms of speed (less number of iterations) and robustness (stable outlier estimation even if the inliers are noisy). Thereby the superiority of ANTSAC over the standard RANSAC algorithm grows with the complexity of the model to be determined and with an increasing relative number of outliers. Moreover, the ANTSAC sampling strategy can compete with other state-of-the-art methods, e.g. PROSAC or LO-RANSAC in real-world problem scenarios in that even less runtime is required to produce a similar number of inlier without performing additional domain specific computations.

## II. RANDOM SAMPLE CONSENSUS

In the following, we give a short recapitulation of the original RANSAC algorithm [1]. Thereby the mathematical notation that will be used in the remaining sections is introduced and some important extensions of RANSAC are discussed shortly.

Let  $S \subseteq \mathbb{S}$  be a finite set of *samples* or *data points*, where  $\mathbb{S}$  denotes the *sample space*. As *model space* we define the set  $\mathbb{M} := \{M \mid M \in \mathcal{P}(\mathbb{S}) \wedge |M| = k\}$  of all possible *model instances* or *hypotheses*  $M$ . Thereby  $k \geq 1$  denotes the number of samples, which are at least required to instantiate a model. We assume that  $S$  contains a subset  $S_{in}^*$  containing more than  $k$  samples that fit a model instance within a prescribed accuracy. Now, let  $d : \mathbb{M} \times \mathbb{S} \rightarrow \mathbb{R}$  be a function, which measures the distance between model instances and samples. With a given problem specific threshold  $\theta \in \mathbb{R}^+$  we define the *set of inliers* or the *consensus set* as

$$S_{in}(M) := \{s \mid s \in S \wedge |d(M, s)| \leq \theta\}. \quad (1)$$

Thereby  $\theta$  defines a boundary, which distinguishes sharply between correct samples (inliers) and incorrect samples (outliers).

The aim of RANSAC is to find the optimal model instance  $M^*$ , that maximizes the consensus set and therefore is the best representative of the given sample set  $S$ . This is achieved by iteratively picking model instances  $M^t$  and comparing the number of inliers i.e.  $|S_{in}(M^t)|$ . Thereby each model instance is chosen with the same probability. In the following, we denote  $s \sim \mathcal{U}(S)$  for choosing a sample  $s$  from the uniformly distributed sample space  $S$ . The model instance  $M^{max}$  that generates the set  $S_{in}^{max} := S_{in}(M^{max})$  with the maximum number of inliers, i.e.  $\forall t : |S_{in}^{max}| \geq |S_{in}(M^t)|$  is the instance we are searching for. Note that the “goodness” of a model instance depends on  $\theta$  and thus a “best-fitting” instance must not necessarily be unique. It only holds  $M^* \equiv_{\theta} M^{max}$ , which means that the searched optimal model  $M^*$  is covered by the consensus set of  $M^{max}$  under the threshold  $\theta$ . In conclusion, the complete RANSAC procedure is depicted in Algorithm 1.

---

### Algorithm 1 The original RANSAC algorithm

---

```

1:  $S_{in}^{max} := \{ \}$ 
2: for  $t := 1$  to  $T$  do
3:   select  $s_1^t, \dots, s_k^t \sim \mathcal{U}(S)$ 
4:    $M^t := \text{build\_model}(s_1^t, \dots, s_k^t)$ 
5:   determine  $S_{in}^t$  using eq. (1)
6:   if  $|S_{in}^t| > |S_{in}^{max}|$  then
7:      $S_{in}^{max} := S_{in}^t$ 
8:      $M^{max} := M^t$ 
9:   end if
10: end for
11: return  $(S_{in}^{max}, M^{max})$ 

```

---

Meanwhile the RANSAC algorithm has been modified in many different ways. There is an almost overwhelming amount of publications available concerning the improvement of the RANSAC algorithm and it is out of the scope of this work to mention all of them. As proposed in [3], the various variants of RANSAC can be divided into three categories yielding one of the objectives accuracy, speed, and robustness.

Accuracy is e.g. addressed by the maximum a posteriori estimator MAPSAC [4] which is based on the maximum likelihood estimation sample consensus (MLESC) proposed by Torr and Zisserman [5]. Thereby MLESC is quite similar to MSAC (m-estimator sample consensus) (see also [5]) but could be shown to provide a slightly lower estimation error. Other approaches addressing accuracy are the robust projection based M-estimator by [6], the LO-RANSAC algorithm [7] which adds a generalized model optimization step to the original RANSAC algorithm, or QDEGSAC (RANSAC algorithm for (quasi-)degenerate data) presented by Frahm and Pollefeys [8].

Speed is e.g. addressed by the usage of priors in guided-MLESC [9], by restricting the set of data from which samples are drawn in PROSAC [10], by selecting adjacent data points as samples in NAPSAC [11], by combining RANSAC with evolutionary optimization techniques in GASAC [12], by various randomized RANSAC variants (e.g. [13], [14]), or by a preemptive scoring mechanism [15]. Other recent approaches addressing fastness are e.g. the optimal randomized RANSAC proposed in [16], SCRAMSAC [17], Adaptive Sample Consensus [18], or deterministic sample consensus (DESAC) [19].

Robustness is e.g. addressed by using random minimum subsets to find inliers (FH-MAPSAC) [20], by estimating the outliers share as well as the inliers noise level (AM-LESAC) [21], or by expectation maximization (u-MLESC) [22]. Conclusively, there are many RANSAC extensions that provide significant enhancements in terms of either robustness, speed or quality. Many other methods were introduced, e.g. SimSAC and BaySAC [23] having pros and cons. BaySAC halve the number of iterations compared to RANSAC but require a prior estimation of inlier probabilities, while SimSAC is computational expensive. Another example is the USAC algorithm [24], which is not an entirely new algorithm but rather a combination of various existing methods, such as PROSAC, LO-RANSAC and others.

Nonetheless, the probably most convincing aspect of the standard RANSAC method, and arguably the reason why it is still commonly used, is its simplicity. Unfortunately, up to now there exists no method that is similarly simple but also provides good improvements by keeping the computational overhead small, such that it would be attractive for a broad range of practical applications. In the remainder of this paper we outline an approach that will fulfill all of these requirements.

## III. ANTSAC APPROACH

As mentioned above our approach is based on a mechanism that memorizes the samples that have been approved to fit the searched model instance  $M^*$ . Thereby, to fit the model instance means that a sample is contained in the consensus set. Without any further knowledge of the model instance or sample distribution, the most intuitive way to measure the probabon of a sample is to memorize whenever it is part of a consensus set or not over the whole iteration process. Under the assumptions from the previous section the result is, that samples, which are part of the optimal model  $M^*$ , are more often determined as inliers than samples which are indeed outliers. For shorthand, we will refer to this effect as *tendential inlier probabon* (TIP).

In Section II we mentioned that the decision of a sample to be an inlier or not is sharp and depends on the threshold  $\theta$ . Even if  $\theta$  is chosen carefully this might cut off information which could be used additionally for the TIP in a similar manner as the binary information to be an inlier or not.

As the probability of finding  $M^*$  using the RANSAC algorithm rises with the number of iterations, the TIP becomes more obvious, but is already noticeable after just a few iterations. To exploit this information during the iterative process we need an appropriate memory to store the TIP information for each sample. Additionally, we also have to modify the selection process in such a way that the TIP is respected profitably. In our approach both requirements are solved through techniques based on the *Ant Colony Optimization* (ACO) algorithm [2].

The ACO meta-heuristic, inspired by the pheromone based organization of foraging ants, was successfully applied to many combinatorial optimization tasks as well as numerical optimization problems. Thereby a pheromone matrix is used that stores a pheromone level for each particular solution. The pheromone levels vaporize over time and are refreshed iteratively by artificial ants proportional to the fitness of each superordinate entire solution. Which particular solution is chosen locally by an ant depends on a probability distribution based on the current pheromone matrix as well as some additional local heuristic information. The higher the pheromone level of a possible particular solution is, the more likely it is getting picked.

In order to use the aspects of ACO to utilize the effect of emergence for outlier estimation we introduce the pheromone memory as a function  $\tau : S \rightarrow \mathbb{R}$  that assigns a pheromone value to each sample of  $S$ . Thereby the convention is that the higher the pheromone level of a specific sample is, the higher its importance is related to the other samples. Before the first iteration, this memory is initialized for all samples through

$$\tau^1(s) := \frac{1}{|S|}. \quad (2)$$

After each iteration of the ANTSAC algorithm the pheromone memory  $\tau$  is updated by

$$\tau^{t+1}(s) := \rho\tau^t(s) + \Delta\tau^t(s). \quad (3)$$

The rate  $\rho \in [0, 1]$  defines the degree of the pheromone level evaporation. If  $\rho$  is chosen too small then information is “washed” out of the memory very fast, whereby “good” information may get lost. Contrarily, if  $\rho$  is chosen too large, the process of emergence might be disrupted by too much “bad” information accumulated in the memory. Evaporation rates around 0.9 are proposed to be appropriate (see e.g. [25]). The offset  $\Delta\tau^t(s)$  gives the pheromone refreshment after iteration  $t$ . We propose the refreshment to be

$$\Delta\tau^t(s) := \frac{|S_{in}^t|}{|S| + \frac{1}{t} \sum_{t' \leq t} |S_{in}^{t'}|} \exp\left(-\frac{1}{2} \left[\frac{d(M^t, s)}{\theta}\right]^2\right) \quad (4)$$

The first part of (4) is the basis rewarding expressing the goodness of the current model instance. The rewarding is higher the greater the number of inliers in the current iteration

is. Additionally, the mean number of inliers achieved so far is used to damp the rewarding if there is no more improvement. The second part of equation (4) scales the rewarding related to the distance of each sample to the current model instance. Thereby, the scale is computed with a gaussian-like function, where the maximum is at 0 and the inflection point is at  $\theta$ .

Now, the current pheromone level for each sample can be expressed as the probability

$$P(s; t) := \frac{\tau^t(s)^\alpha}{\sum_{s' \in S} \tau^t(s')^\alpha}, \quad (5)$$

where  $\alpha \in \mathbb{R}^+$ , usually selected in the interval  $[0.1, 2.0]$ , adjusts the relative importance of each sample. The higher  $\alpha$  is the more high pheromone values are strengthened and low pheromone values are attenuated. With  $\mathcal{D}(S, t)$ , we further denote the probability distribution over  $S$  at iteration  $t$  holding (5) for each sample of  $S$ . Thus  $s \sim \mathcal{D}(S, t)$  means that a sample  $s$  is selected from  $S$  corresponding to that given probability distribution. Compared to common ACO implementations, we use no objective heuristic in equation (5) such that the probability of a sample to be picked depends only on its relative pheromone level. Introducing a domain specific heuristic might be very useful, but has not been tested so far.

Combining a pheromone memory and probabilistic picking, the ANTSAC algorithm no longer is a pure *Monte Carlo* method like classic RANSAC, but rather a meta-heuristic search strategy, where the TIP gives some kind of search direction. Thereby, our method provides the real inlier candidates to emerge from the set samples over time (see Figure 1). This phenomenon is discussed and illustrated in Section V. The complete ANTSAC procedure is summarized in Algorithm 2.

---

#### Algorithm 2 The ANTSAC algorithm

---

```

1:  $S_{in}^{max} := \{ \}$ 
2: for all  $s \in S$  do
3:    $\tau^1(s) := \frac{1}{|S|}$ 
4: end for
5: for  $t = 1$  to  $T$  do
6:   select  $s_1^t, \dots, s_k^t \sim \mathcal{D}(S, t)$ 
7:    $M^t := \text{build\_model}(s_1^t, \dots, s_k^t)$ 
8:   determine  $S_{in}^t$  using Eq. (1)
9:   if  $|S_{in}^t| > |S_{in}^{max}|$  then
10:     $S_{in}^{max} := S_{in}^t$ 
11:     $M^{max} := M^t$ 
12:   end if
13:   for all  $s \in S$  do
14:     determine  $\Delta\tau^t(s)$  using Eq. (4)
15:      $\tau^{t+1}(s) := \rho\tau^t(s) + \Delta\tau^t(s)$ 
16:   end for
17: end for
18: return  $(S_{in}^{max}, M^{max})$ ;

```

---

## IV. EXPERIMENTS

The aim of our experiments is mainly twofold. First, we analyze the ANTSAC algorithm and its improvement compared to vanilla RANSAC on three different artificial problems, namely, line fitting (2 data points/model, referred as LINE),

ellipse fitting (5 data points/model, referred as ELLIPSE) and fundamental matrix estimation (8 data points/model, referred as FUNDAMENTAL). Here, the task is to study the behavior of ANTSAC and the influence of its parameters on different complex problem scenarios. Thereby, each particular problem is performed with four different values for  $q$  (0.5, 0.4, 0.3 and 0.2) that refers to the relative number of inliers.

Second, we compared ANTSAC with various state-of-the-art methods using the USAC framework presented in [24] on real-world data for homography estimation (4 data points/model, referred as HOMOGRAPHY) and essential matrix estimation (5 data points/model, referred as ESSENTIAL). Since ANTSAC is, strictly speaking, only a sampling strategy without an additional optimization step or a special stopping criterion, its nearest competitor is PROSAC [10]. However, we compared it with LO-RANSAC [7] and USAC-1.0 [24] as well. In order to get meaningful results all these experiments was repeated 500 times.

The general performance analysis of ANTSAC bases on the following terms. Let the *average performance curve*  $f$  for  $X$  repetitions be defined by

$$f_A(t) := \frac{1}{X} \sum_{x=1}^X |S_{in,x}^t|. \quad (6)$$

This curve gives the average number of inliers produced by a RANSAC-like algorithm  $A$  after  $t$  iterations. The *improvement curve* gives the relative performance of an ANTSAC instance compared to vanilla RANSAC and is computed with

$$g_{ANTSAC(\alpha,\rho)}(t) := \frac{f_{ANTSAC(\alpha,\rho)}(t)}{f_{RANSAC}(t)}. \quad (7)$$

Thus,  $g_{ANTSAC(\alpha,\rho)}(t) > 1$  means that the given ANTSAC instance produces more inliers than RANSAC after  $t$  iterations. In this context the term *maximum improvement* refers to  $\max_t \{g_{ANTSAC(\alpha,\rho)}(t)\}$ .

To evaluate which  $\alpha, \rho$  combinations work best, we process as follows. For each particular problem and the four different values of  $q$  various experiments were performed covering a sufficiently dense map of parameter configurations over  $\alpha \in [0.1, 2.3]$  and  $\rho \in [0.1, 1.0]$ . Then, for each such parameter configuration the improvement curve over 500 repeats was computed. With these improvement curves an *improvement map*  $G(\alpha, \rho)$  was set up, which contains the improvements either after a fixed number of iterations or the maximal improvements. Finally, we determined the relative goodness of a parameter pair through

$$\hat{G}(\alpha, \rho) := \frac{G(\alpha, \rho) - \min_{\forall \alpha', \rho'} G(\alpha', \rho')}{\max_{\forall \alpha', \rho'} G(\alpha', \rho') - \min_{\forall \alpha', \rho'} G(\alpha', \rho')}. \quad (8)$$

In [24] for PROSAC sampling the samples are given in a priori sorted order according to their putative quality. We found that a good way to acquire this information in ANTSAC is to scale the initial pheromone values of the ordered samples using a function that declines like the density function of a normal distribution. Therefore we propose

$$\tau^1(s) := \frac{1}{|S|} \left[ \lambda + (1 - \lambda) \exp \left( -\frac{1}{2} \left[ \frac{\pi(s)}{\sigma|S|} \right]^2 \right) \right]. \quad (9)$$

Here  $\lambda \in [0, 1]$  is a minimal value to which the scaling term converges (we fixed  $\lambda$  to  $|S|^{-1}$ ),  $\pi : S \rightarrow \{0, \dots, |S| - 1\}$  models a permutation of  $S$  representing the putative quality ordering and  $\sigma \in [0, 1]$  defines the inflection point of the scaling term (we fixed  $\sigma$  to 0.1). To distinguish results for ANTSAC with or without a priory pheromone map initialization we refer the latter here as pk-ANTSAC (prior knowledge).

## V. RESULTS AND DISCUSSION

Figure 2 visualizes the relative goodness for various parameter configurations for our three toy problems. More precisely, each plot shows the average relative goodness as computed in equation (8) over all four  $q$ 's. It can be observed that for each particular problem there is an unique area on the  $\alpha, \rho$ -map for which ANTSAC works best. This area is already noticeable even in the early phase (upper row of Figure 2). When considering the goodness for the maximal improvement (lower row of Figure 2) these areas come out clearly and appear to be smooth and uni-modal. For LINE the optimum is approximately at  $\alpha = 1.6, \rho = 0.85$ , for ELLIPSE at  $\alpha = 1.3, \rho = 0.95$  and for FUNDAMENTAL at  $\alpha = 1.3, \rho = 0.75$ . An interesting observation is that for FUNDAMENTAL the range of good working parameters is significantly broader than for ELLIPSE but also for LINE. However, parameter pairs at roughly about  $\alpha = 1.2-1.4, \rho = 0.9$  seems to work universally.

The performance achieved in our experiments is depicted in Figure 3. Already in the early phase, after just 5 iterations for LINE, 10 for ELLIPSE and 15 for FUNDAMENTAL there is on average a considerable improvement for all three problems, except the hard experiments with a low inlier relation. But after approximately 10 iterations for LINE, 20 for ELLIPSE and 30 for FUNDAMENTAL, ANTSAC can unfold its potential as a meta-heuristic search strategy. Thus, the maximal improvements rise up to 16–25% for LINE, 35–39% for ELLIPSE and 68–160%. It can be seen that, e.g., for the FUNDAMENTAL experiment, the mean inlier number achieved by ANTSAC within 50 iterations is not achieved by RANSAC even after 10,000 iterations. This saves computing hypotheses of several magnitudes and is produced without any domain specific optimization or prior knowledge. Furthermore, the diagrams in Figure 3 offer the probably most important aspect of the behavior of our algorithm: The more difficult the problem becomes the bigger also becomes the performance gain of ANTSAC.

The results regarding the competitiveness of ANTSAC are given in Table I and II. For HOMOGRAPHY it can be seen, that within 100 iterations ANTSAC is able to produce a similar number of inliers than LO-RANSAC and USAC-1.0 using slightly less run-time. However, it significantly outperforms PROSAC sampling. For ESSENTIAL pk-ANTSAC outperforms PROSAC already in the early phase in more than 8 times less run-time. Within 100 iterations, the maximum inlier number of LO-RANSAC and USAC is not achieved, but the result of ANTSAC as well as pk-ANTSAC is way better than PROSAC. Conclusively, the important outcome here is that ANTSAC can compete on real-world problems with domain specific optimized algorithms and performs even better than the state-of-the-art sampling strategy PROSAC that utilizes prior knowledge. It seems possible that ANTSAC may inspire

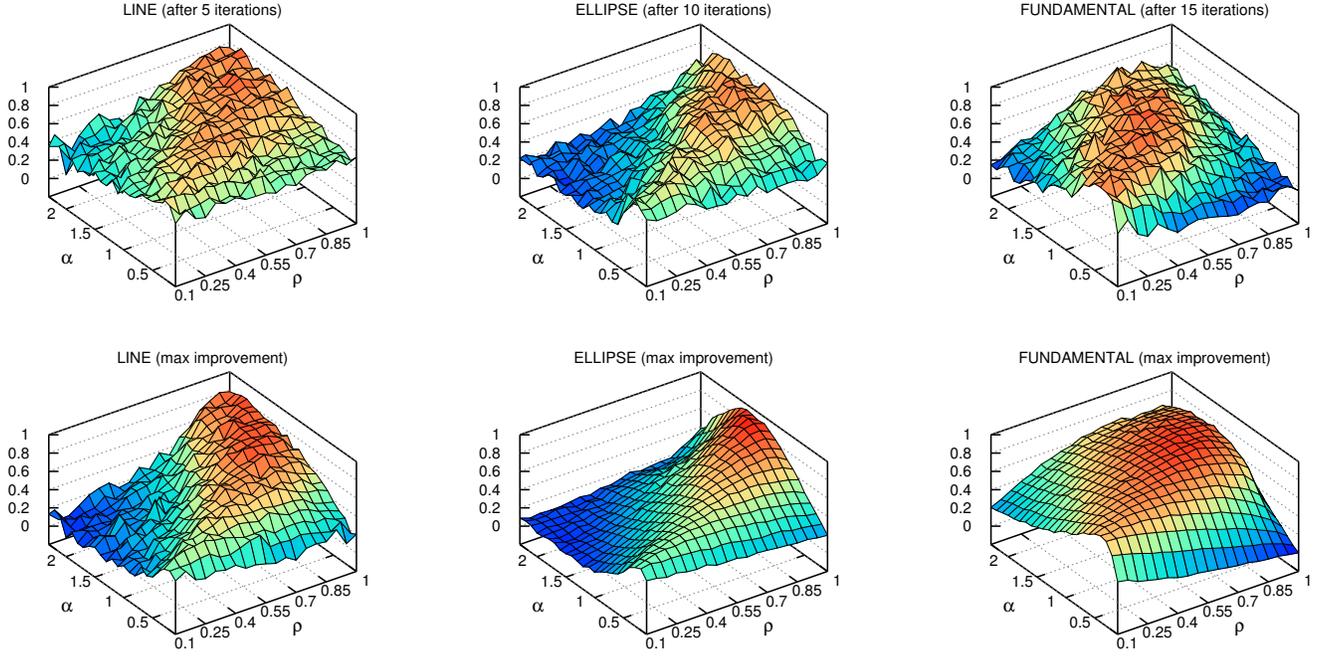


Fig. 2. Visualization of the influence of the two major ANTSAC parameters  $\alpha$  and  $\rho$  to the performance of ANTSAC for line and ellipse fitting as well as fundamental matrix estimation. The upper row shows the achieved performance in a very early phase, while the lower row shows the maximum achieved performance. The higher the value at a certain coordinate is, the better ANTSAC performs with the corresponding parameter pair.

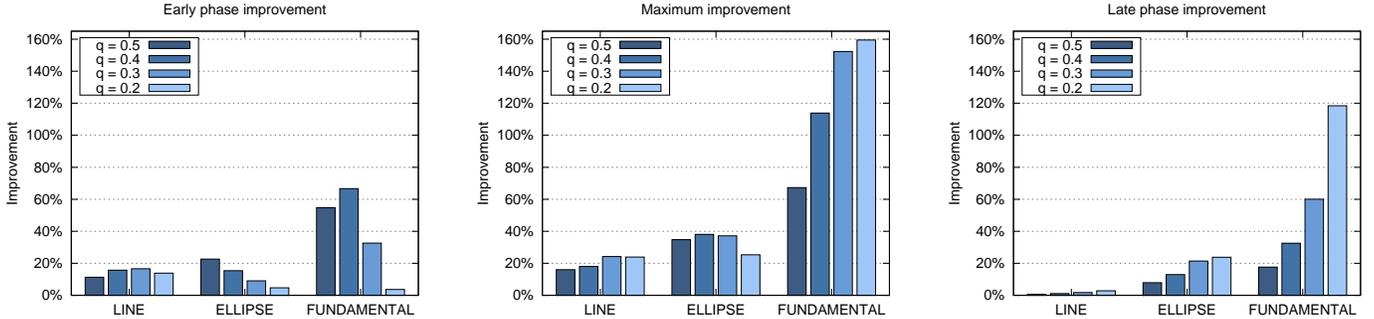


Fig. 3. The improvement of ANTSAC compared to vanilla RANSAC regarding the achieved number of inliers after a certain number of iterations for different inlier relations. The left diagram shows the improvement in the early phase of the algorithm, the center diagram depicts the maximal achieved improvement and the right diagram shows the improvement after 500 iterations for LINE and after 1000 iterations for ELLIPSE and FUND.

TABLE I. ANTSAC COMPARISON FOR HOMOGRAPHY

Hypo. limit	Algorithm	Hypo. used	Inliers	Runtime
10	PROSAC	9	839.93 $\pm$ 221.80	42 [ms]
	LO-RANSAC	10	989.74 $\pm$ 361.30	31 [ms]
	USAC-1.0	6	1 131.85 $\pm$ 135.01	80 [ms]
	ANTSAC	10	488.12 $\pm$ 325.33	5 [ms]
	pk-ANTSAC	10	727.15 $\pm$ 226.73	5 [ms]
100	PROSAC	16	978.25 $\pm$ 114.45	43 [ms]
	LO-RANSAC	100	1 130.45 $\pm$ 52.09	50 [ms]
	USAC-1.0	6	1 147.55 $\pm$ 2.73	81 [ms]
	ANTSAC	100	1 049.41 $\pm$ 55.72	47 [ms]
	pk-ANTSAC	100	1 054.19 $\pm$ 52.38	47 [ms]

TABLE II. ANTSAC COMPARISON FOR ESSENTIAL

Hypo. limit	Algorithm	Hypo. used	Inliers	Runtime
10	PROSAC	10	361.86 $\pm$ 49.54	41 [ms]
	LO-RANSAC	10	560.50 $\pm$ 150.26	29 [ms]
	USAC-1.0	10	628.94 $\pm$ 248.28	73 [ms]
	ANTSAC	10	306.43 $\pm$ 138.06	5 [ms]
	pk-ANTSAC	10	383.98 $\pm$ 114.78	5 [ms]
100	PROSAC	33	458.10 $\pm$ 80.12	45 [ms]
	LO-RANSAC	100	661.05 $\pm$ 87.95	55 [ms]
	USAC-1.0	65	722.86 $\pm$ 52.80	81 [ms]
	ANTSAC	100	552.47 $\pm$ 60.16	47 [ms]
	pk-ANTSAC	100	568.76 $\pm$ 62.18	48 [ms]

synergistic in combination with other techniques, e.g., LO-RANSAC. Last, it should be noted that the shrinking gap between ANTSAC and pk-ANTSAC over time indicates the generic knowledge acquisition ability of ANTSAC sampling.

## VI. CONCLUSION AND FUTURE WORK

We have presented a novel approach for model fitting combining the well-known RANSAC method with principles of ant colony algorithms. The resulting ANTSAC algorithm is

equipped with a pheromone memory, which gains information from the process of iteratively picking model instances, while the picking itself becomes more and more intelligent over time. Thereby, ANTSAC provides the real inlier candidates to emerge from the set of samples.

We could show that ANTSAC dramatically outperforms classical RANSAC both in terms of smaller number of iterations needed and also in robustness. For instance, in the case of fundamental matrix estimation ANTSAC produces up to 160 % more inliers than RANSAC. Further, to gain the same number of inliers ANTSAC needs only a fractional part of iterations. Actually, it turned out that ANTSAC often finds optimal model instances within 50 iterations that RANSAC does not find even after more than 10 000 iterations. It also found that the improvement of ANTSAC grows with the difficulty of the problem. This is interesting because in the context of a more complex problem on one hand instantiating the model and determining distances becomes computationally more expensive and on the other hand the number of recommended RANSAC-iterations nearly explodes.

While other RANSAC variants require domain specific knowledge of a problem, which produces a lot of computational overhead or is difficult to implement, the ANTSAC algorithm as it is presented in this paper is totally generic, performs efficiently, and is very simple to implement. In fact, it is strikingly easy to integrate ANTSAC sampling into an existing RANSAC implementation. However, we figured out that ANTSAC as a generic approach is competitive even to domain specific optimized state-of-the-art methods like, e.g. PROSAC or LO-RANSAC.

One future task is to evaluate the final fitting capability of our algorithm. Usually, after performing RANSAC or similar methods, the entire consensus set is used to estimate a final model instance, e.g., a least-squares solution. We think that the final pheromone matrix snapshot after performing ANTSAC might contain enough information, namely an additional weighting of the consensus set, to derive the final best-fitting-model more or less directly. Another research task is to investigate if ANTSAC is suited for dynamic and multimodal problem environments. It could also be interesting to derive a stopping criterion from the distribution of the pheromone map, e.g. by checking if there is a stable state reached.

## REFERENCES

- [1] M. Fischler and R. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [2] M. Dorigo and G. D. Caro, "The ant colony optimization meta-heuristic," in *New Ideas in Optimization*. McGraw-Hill, 1999, pp. 11–32.
- [3] S. Choi, T. Kim, and W. Yu, "Performance evaluation of RANSAC family," in *BMVC*. British Machine Vision Association, 2009.
- [4] P. H. S. Torr, "Bayesian model estimation and selection for epipolar geometry and generic manifold fitting," *International Journal of Computer Vision*, vol. 50, no. 1, pp. 35–61, Oct. 2002.
- [5] P. Torr and A. Zisserman, "MLESAC: a new robust estimator with application to estimating image geometry," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 138–156, Apr. 2000.
- [6] R. Subbarao and P. Meer, "Subspace estimation using projection based m-estimators over grassmann manifolds," in *Computer Vision ECCV 2006*, ser. Lecture Notes in Computer Science, A. Leonardis, H. Bischof, and A. Pinz, Eds. Springer Berlin Heidelberg, Jan. 2006, no. 3951, pp. 301–312.
- [7] O. Chum, J. Matas, and S. Obdrzalek, "Enhancing RANSAC by generalized model optimization," in *Proc. of the Asian Conference on Computer Vision (ACCV)*, K.-S. Hong and Z. Zhang, Eds., vol. 2. Seoul, Korea South: Asian Federation of Computer Vision Societies, Jan. 2004, pp. 812–817.
- [8] J.-M. Frahm and M. Pollefeys, "RANSAC for (quasi-)Degenerate data (QDEGSAC)," in *Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition - Volume 1*, ser. CVPR '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 453–460.
- [9] B. Tordoff and D. Murray, "Guided-MLESAC: faster image transform estimation by using matching priors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1523–1535, Oct. 2005.
- [10] O. Chum and J. Matas, "Matching with PROSAC - progressive sample consensus," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005*, vol. 1, Jun. 2005, pp. 220–226.
- [11] D. R. Myatt, P. H. S. Torr, S. J. Nasuto, J. M. Bishop, and R. Craddock, "NAPSAC: high noise, high dimensional robust estimation - it's in the bag," in *BMVC*, P. L. Rosin and A. D. Marshall, Eds. British Machine Vision Association, 2002.
- [12] V. Rodehorst and O. Hellwich, "Genetic algorithm SAmple consensus (GASAC) - a parallel strategy for robust parameter estimation," in *Conference on Computer Vision and Pattern Recognition Workshop, 2006. CVPRW '06*, Jun. 2006, pp. 103–103.
- [13] J. Matas and O. Chum, "Randomized RANSAC with  $t_{d,d}$  test," *Image and Vision Computing*, vol. 22, no. 10, pp. 837–842, Sep. 2004.
- [14] —, "Randomized RANSAC with sequential probability ratio test," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2, Oct. 2005, pp. 1727–1732.
- [15] D. Nister, "Preemptive RANSAC for live structure and motion estimation," in *Ninth IEEE International Conference on Computer Vision, 2003. Proceedings*, vol. 1, Oct. 2003, pp. 199–206.
- [16] O. Chum and J. Matas, "Optimal randomized RANSAC," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 8, pp. 1472–1482, Aug. 2008.
- [17] T. Sattler, B. Leibe, and L. Kobbelt, "SCRAMSAC: improving RANSAC's efficiency with a spatial consistency filter," in *Computer Vision, 2009 IEEE 12th International Conference on*, Oct. 2009, pp. 2090–2097.
- [18] L. Fan and T. Pylvinenen, "Adaptive sample consensus for efficient random optimization," in *Advances in Visual Computing*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, Jan. 2009, no. 5876, pp. 252–263.
- [19] P. McIlroy, E. Rosten, S. Taylor, and T. Drummond, "Deterministic sample consensus with multiple match hypotheses," in *Proceedings of the British Machine Vision Conference*. BMVA Press, 2010, pp. 111.1–111.11.
- [20] C. L. Feng and Y. S. Hung, "A robust method for estimating the fundamental matrix," in *In International Conference on Digital Image Computing*, 2003, pp. 633–642.
- [21] A. Konushin and V. Gaganov, "AMLESAC: a new maximum likelihood robust estimator," in *GraphiCon*, Novosibirsk Akademgorodok, Russia, 2005.
- [22] S. Choi and J.-H. Kim, "Robust regression to varying data distribution and its application to landmark-based localization," in *IEEE International Conference on Systems, Man and Cybernetics, 2008. SMC 2008*, Oct. 2008, pp. 3465–3470.
- [23] T. Botterill, S. Mills, and R. D. Green, "New conditional sampling strategies for speeded-up ransac," in *BMVC*. British Machine Vision Association, 2009.
- [24] R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J.-M. Frahm, "USAC: a universal framework for random sample consensus," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 2022–2038, Aug. 2013.
- [25] E. Bonabeau, G. Theraulaz, and M. Dorigo, *Swarm Intelligence: From Natural to Artificial Systems (Santa Fe Institute Studies in the Sciences of Complexity)*. Oxford University Press, USA, 1999.