BetaSAC: A New Conditional Sampling For RANSAC

Antoine Meler, Marion Decrouez, James L. Crowley

BetaSAC: A New Conditional Sampling For RANSAC

Antoine Meler[1]
antoine.meler@inrialpes.fr
Marion Decrouez[1,2]
marion.decrouez@inrialpes.fr
James L. Crowley[1]
james.crowley@inrialpes.fr

[1] INRIA Rhone-Alpes Research Centre
LIG laboratory
Grenoble, France

[2] CEA LIST
Embedded Vision Systems
Saclay, France

Abstract

We present a new strategy for RANSAC sampling named BetaSAC, in reference to the beta distribution. Our proposed sampler builds a hypothesis set incrementally, selecting data points conditional on the previous data selected for the set. Such a sampling is shown to provide more suitable samples in terms of inlier ratio but also of consistency and potential to lead to an accurate parameters estimation. The algorithm is presented as a general framework, easily implemented and able to exploit any kind of prior information on the potential of a sample. As with PROSAC, BetaSAC converges towards RANSAC in the worst case. The benefits of the method are demonstrated on the homography estimation problem.

1 Introduction

RANSAC (RANdom Sample Consensus) [4] is a non-deterministic algorithm for the estimation of a mathematical model from observed data which contains outliers. It is essentially composed of two steps that are repeated iteratively.

- **Hypothesize.** A sample of size m among the N data points is randomly selected. The model parameters are computed from this sample. m is the smallest sufficient cardinality to determine the model parameters.

- **Test.** The hypothesis is verified against the rest of the data by counting the points consistent with the estimated model parametrization.

These two phases are repeated until the probability of finding a better solution falls below a pre-selected threshold. The number of necessary iterations increases very fast with the outlier ratio and the model complexity m. Many evolutions of RANSAC have been proposed over the last thirty years. In the following section, we evoke the most important improvements of the hypotheses generation (sampling process). Then, we present our own proposed sampler.

BMVC 2010 doi:10.5244/C.24.42

2 Non-uniform sampling

Numerous methods have been derived from RANSAC. Many of them change the sampling to be faster. In this section, we distinguish between two types of non-uniform sampling: *biased sampling* and *reordered sampling*.

2.1 Biased Sampling

Biasing the sampling can accelerate RANSAC by reducing the necessary number of iterations. The bias is usually based on additional information giving an *a priori* on the correctness of a datum.

Guided-MLESAC [12] uses prior probabilities of the validities of the data. Data points having a high inlier probability are more likely to be selected to form minimal hypothesis sets. Other methods use the results of the past iterations as an additional source of information. This is the case of BaySAC and SimSAC [1] which select the hypothesis set the most likely to be correct, conditional on the knowledge of those that have failed to lead to a good model. NAPSAC (N Adjacent Points SAC [9]) uses heuristics that an inlier tends to be closer to other inliers than outliers.

2.2 Reordered Sampling

A biased sampling has the risk of impairing the search if the information that is used is not a good *a priori* or if the randomization is insufficient. Some methods overcome this risk by limiting the guidance to a reordering of the generated samples. Thus, with a minimum of hypothesis on the information used to guide sample drawing, PROSAC [2] and GroupSAC [10] ensure not to be worse than a random selection. During the first T_N iterations, samples are generated beginning with the most probable and finishing by the least probable. After this phase, each sample has had the same chance of being formed (property $\mathscr{P}_1$), but the samples considered better are more likely to be drawn earlier (property $\mathscr{P}_2$). Then, the sampling is uniform. Choosing T_N as the average number of iterations needed by RANSAC to find a good model ensures an identical worst-case behavior. As RANSAC methods stop as soon as a good enough model is found, this reordered sampling has a good chance to lead to an early stop. In this section, we unify these methods under a common framework, and we introduce our own sampling strategy.

Let $\mathscr{D} = \{d_1, \ldots, d_N\}$ be a set of N data points and m the number of points needed to estimate the model parameters. Equation 1 defines a random variable in $\mathscr{D}$, depending on the forming minimal sample s ($|s| < m$) and the iteration counter t. During the first T_N iterations, the selection is guided by $X(t,s)$. Then, for $t > T_N$, it is uniform, as in standard RANSAC. We write $d_{U_{\{1,\ldots,N\}}}$, the random variable returning d_k with k drawn by the uniform random variable $U_{\{1,\ldots,N\}}$. We deduce in Equation 2, a random variable in the space of minimal samples $\mathscr{D}^m$. To avoid the use of hypergeometric distributions, we make the common assumption that a selection is done with replacement.

$$D(t,s) = \begin{cases} X(t,s), & t \leq T_N \\ d_{U_{\{1,\ldots,N\}}}, & t > T_N \end{cases} \tag{1}$$

$$S(t,m) = \{ s_1 = D(t,\{\emptyset\}), s_2 = D(t,\{s_1\}), \ldots, s_m = D(t,\{s_1,\ldots,s_{m-1}\}) \} \tag{2}$$

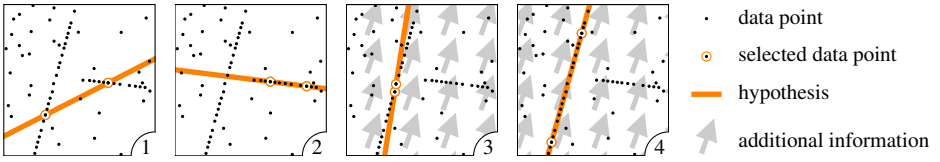


Figure 1: Different types of inlier samples for the line fitting problem. 1) *inlier sample* (all sample points belong to a line). 2) *consistent sample*. 3) *sample consistent with additional information*. 4) *suitable sample*. Thanks to its conditional sampling, BetaSAC is able to generate *suitable samples* earlier than RANSAC and PROSAC would do.

BetaSAC and its inspiring methods [2, 4, 10] all differ essentially by the selection random variable $X(t,s)$. In standard RANSAC (Equation 3), the selection is uniform even for $t \leq T_N$. In PROSAC (Equation 4), $d_{(k)}$ stands for the k^{th} data point in a sorting. This sorting is performed once and for all, based on an inlier prior associated to each data point. $g(t)$ is a growth function which limits a uniform selection in a progressively larger set of top ranked data points. GroupSAC (Equation 5) makes a uniform selection in a configuration defined as a union of predefined data groups $\mathscr{G}(t) = \{G_i\}_{i=1\ldots k}$. As demonstrated by the authors, configurations with fewer groups are more likely to give an all-inlier sample and are therefor examined earlier. Our proposed sampling strategy (Equation 6) strongly depends on the forming sample s. The realization of a random variable, $B_{i_{|s|}(t)/n}$, gives the rank of the selected datum in a sorting depending on s, written $(.)_s$. We will detail in Section 4 the constant n, the function $i_{|s|}(t)$ and the random variable $B_{i_{|s|}(t)/n}$.

$$X_{RANSAC}(t,s) = d_{U_{\{1,\ldots,N\}}} \tag{3}$$

$$X_{PROSAC}(t,s) = \begin{cases} d_{(g(t))} & \text{if } |s| = 0 \\ d_{(U_{\{1,\ldots,g(t)-1\}})} & \text{otherwise} \end{cases} \tag{4}$$

$$X_{GroupSAC}(t,s) = d_{U_{\mathscr{G}(t)}} \tag{5}$$

$$X_{BetaSAC}(t,s) = d_{(B_{i_{|s|}(t)/n})_s} \tag{6}$$

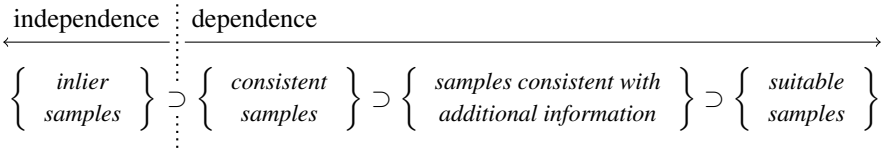
We write $\mathbb{E}$ the expected value of a random variable and $q(s)$ the *a priori* quality of a sample s. Equation 7 is a reformulation of the properties $\mathscr{P}_1$ (each sample has the same chance of being formed) and $\mathscr{P}_2$ (the ones considered better are more likely to be drawn earlier) in the proposed framework.

$$\begin{aligned} \mathscr{P}_1 : \quad & \mathbb{E}\left[\sum_{t=1}^{T_N} (S(t,m) = s)\right] = \frac{T_N}{N^m}, \quad \forall s \in \mathscr{D}^m \\ \mathscr{P}_2 : \quad & t \leq t' \;\Rightarrow\; \mathbb{E}\left[q(S(t,m))\right] \geq \mathbb{E}\left[q(S(t',m))\right] \end{aligned} \tag{7}$$

3 A New Conditional Selection

In this paper, we introduce an approach for a data point selection conditioned on the points previously selected for the current hypothesis set. Contrary to previous methods, BetaSAC does not assume that inlier probabilities of data points are independent. Breaking the independence hypothesis allows the algorithm to generate samples having more potential. We distinguish between four types of samples: *inlier samples*, *consistent samples*, *samples consistent with additional information* and *suitable samples*. The last three require a conditional

sampling to be generated. Figure 1 is a graphical exemplification of them.



Inlier samples are defined as samples containing only inlier data points. Most of the guided sampling methods concentrate on generating *inlier samples*, which do not involve a dependence on the data points. This is what PROSAC [2] does by using the matching score. However, in presence of multiple model parametrizations in the data, an *inlier sample* has a good chance of containing data from different ones, leading to an incorrect hypothesis.

Consistent samples are *inlier samples* passing some consistency constraints. Many kind of such constraints have been studied. For the epipolar geometry estimation problem, many of them come from oriented projective geometry [6, 11, 13]. In a similar manner, Zelnik-Manor *et al*. [14] show that all relative homographies of a pair of planes across multiple views spans an only 4-dimensional linear subspace. Other heuristics can be used as an *a priori* consistency of a sample. NAPSAC (N Adjacent Points SAC) [9] uses the observation that an inlier tends to be closer to other inliers than outliers.

Samples that are *consistent with additional information* have even more potential. A useful and always available additional information in computer vision is the image signal itself. This information is often underexploited while, as shown in Figure 2, it can be very useful for distinguishing between inliers and outliers. GroupSAC [10] demonstrates how a selection depending on a segmentation of the image can be advantageous. Cues on the searched parameters may also come from previous image frames, motion sensors or approximative calibration.

Suitable samples are the type of samples one wants to generate. They are samples able to lead to an accurate estimation of the target parametrization(s). In particular, they must not be degenerate. Many degeneracy criteria have been studied for the problem of epipolar geometry estimation. Such criteria are usually used to reject bad samples *a posteriori* [3, 5]. *Suitable samples* may also be less affected by the inevitable measurement noise. Towards this goal, a sample spanning a big area is often preferable.

BetaSAC offers a conditional sampling which is able to generate more *suitable samples* than pure random would do during the first iterations. The only hypothesis required is that suitable samples can be built by successive data point selections.

4 Algorithm

4.1 The Selection Random Variable

BetaSAC is characterized by its random variable $X_{BetaSAC}(t,s)$ defined in Equation 6, where s is the partial minimal hypothesis set, being built at iteration t. $X_{BetaSAC}(t,s)$ is the result of the selection of the k^{th} data point in a sorting depending on s, where k is a value in $\{1,\ldots,N\}$ drawn by the random variable $B_{i_l(t)/n}$. In previous section, we have presented many cues usable to define a ranking of the data points with respect to a given sample s. In this section, we define the random variable $B_{i_l(t)/n}$.

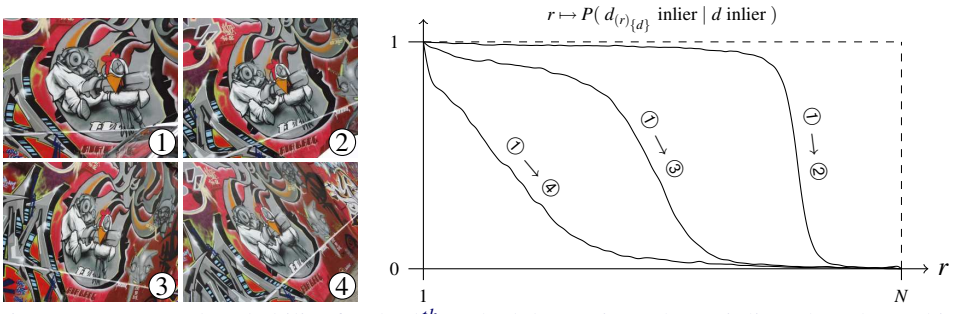


Figure 2: Measured probability for the k^{th} ranked data point to be an inlier when the ranking is done with respect to a sample made of one inlier point. The task is homography estimation and the ranking criterion is the distance between affine matrices associated to the correspondences (see Equation 17 for details). We use the affine invariant feature detector of Mikolajczyk and Schmid [8] and the SIFT descriptor [7]. Left: compared images. Right: plots resulting of the 3 comparisons ① → ②, ① → ③ and ① → ④. One can observe that top ranked correspondences are much more likely to be inliers.

$B_{i_l(t)/n}$ must allow to pick a data point in a desired region of the sorting. The first iterations must draw top ranked data points while respecting $\mathscr{P}_1$ which ensure that each sample has had the same chance of being drawn after T_N iterations. Finding such random variables is easy. We retained a subfamily of the beta probability law. The general beta distribution of shape parameters (α, β) is defined by the density function of Equation 8, with $\text{Beta}(\alpha, \beta)$ the beta function defined in Equation 9.

$$f_{Beta}(x;\ \alpha,\ \beta) = \frac{1}{\text{Beta}(\alpha,\beta)}\, x^{\alpha-1}(1-x)^{\beta-1} \tag{8}$$

$$\text{Beta}(\alpha,\beta) = \int_0^1 t^{\alpha-1}(1-t)^{\beta-1}\, dt \tag{9}$$

The probability density function $f_{B_{i_l(t)/n}}$ of our random variable is defined from the beta distribution on Equation 10 and plotted Figure 3.

$$f_{B_{i_l(t)/n}}(x) = \frac{1}{N-1} f_{Beta}(\frac{x-1}{N-1}\ ;\ i_l(t)\ ,\ n - i_l(t) + 1\ ) \tag{10}$$

The parameter n is a constant in $\mathbb{N}^*$ and $i_l(t)$ moves in $\{1,\ldots,n\}$ over the iterations. Their values will be discussed in Section 4.2.

The reasons why we chose the beta distribution are multiple. The first and most important is that it does not require a complete sorting of the data for a given sample s. This rare property makes the additional computational cost of our sampling procedure negligible. Furthermore, it is trivial to simulate and makes $\mathscr{P}_1$ and $\mathscr{P}_2$ easy to reach.

Simulation of the random variable. The beta distribution is the distribution of the order statistics of a random sample from the uniform distribution. That makes it easy to simulate. Given an iteration count t, a draw of a complete sample s of size m with our conditional random variable is presented in Algorithm 1.

With the use of a linear selection algorithm for step 5, the computational complexity of this sampling procedure is only $O(m.n)$.

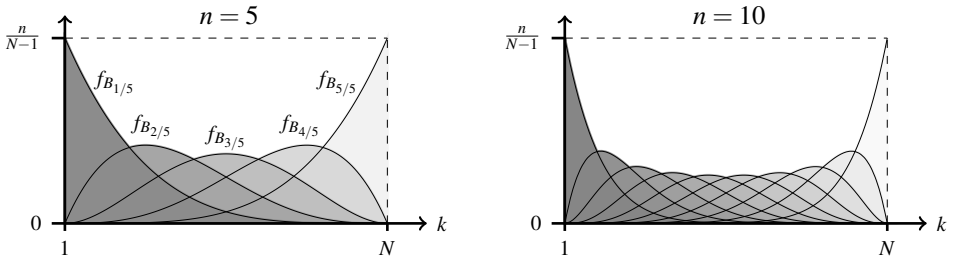


Figure 3: Probability density functions of the n random variables $B_{i/n}, i \in \{1,\dots,n\}$ for $n = 5$ (left) and $n = 10$ (right). Darkest densities correspond to lowest values of i. With a good ranking criterion, the first data points are the more likely to be inliers. Thus, $B_{i/n}$ selects more often inliers than pure random would do for small values of i.

Algorithm 1 Complete sample generation (one draw by $S(t,m)$)

$s \leftarrow \{\emptyset\}$
for $l = 0$ to $m-1$ **do**
  Select n data points at pure random
  Rank the n data points with respect to s
  Find d, the $i_l(t)^{th}$ among the n data points in the ranking
  $s \leftarrow s \cup d$
end for

4.2 The Sampling Strategy

Each BetaSAC iteration starts with the computation of a selection vector $[i_0(t),\dots,i_{m-1}(t)]$, as a function of the iteration counter t. We call these vectors the sampling strategy. In this section, we design the sampling strategy in such a way that properties $\mathscr{P}_1$ and $\mathscr{P}_2$ of Equation 7 are satisfied. As the randomization is assured by $B_{i_l(t)/n}$ even for deterministic values of $i_l(t)$, $i_l(t)$ can be randomized or not. However, we consider the general case of a discrete random variable in $\{1,\dots,n\}$.

Because the density of the uniform selection $f_{U_{[1,N]}}$ is identifiable with our random variable as follows:

$$\frac{1}{n}\sum_{i=1}^{n} f_{B_{i/n}} = f_{U_{[1,N]}} \tag{11}$$

we have convenient sufficient conditions *(a)* and *(b)* on $i_l(t)$ to ensure $\mathscr{P}_1$:

$$\forall (j,j') \in \{1,\dots,n\}^2,\ \forall (l,l') \in \{0,\dots,m-1\}^2,$$

$$(a) \qquad \sum_{t=1}^{T_N} P(i_l(t)=j) = \frac{T_N}{n} \qquad (\textit{equiprobability})$$

$$(b) \quad l \neq l' \ \Rightarrow \ \sum_{t=1}^{T_N} P(i_l(t)=j)).P(i_{l'}(t)=j')) = \frac{T_N}{n^2} \qquad (\textit{uncorrelation}) \tag{12}$$

$\mathscr{P}_2$ involve a sample quality function q. There exists no general optimal function without stronger hypothesis on the s-conditioned sorting $(.)_s$. But we present a family of functions

$q_p, p \in \mathbb{N}^*$ which are a natural generalization of PROSAC *a priori* sample quality. Let $\vec{r}_s = [r_1, \ldots, r_m]$ be the vector of the ranks of the selected data points in a sample s of size m ($\vec{r}_s$ is the m realizations of the random variables $B_{i_0(t)/n}, \ldots, B_{i_{m-1}(t)/n}$):

$$s = \{s_1 = d_{(r_1)_{\{\emptyset\}}}, s_2 = d_{(r_2)_{\{s_1\}}}, \ldots, s_m = d_{(r_m)_{\{s_1,\ldots,s_{m-1}\}}}\}$$

The quality function q_p is defined as the opposite of the *p-norm* of the ranks vector:

$$q_p(s) = -\|\vec{r}_s\|_p = -\sqrt[p]{\sum_{l=1}^{m} r_l^p} \tag{13}$$

This definition of q has the advantage of allowing a reformulation of $\mathscr{P}_2$ in terms of sum of the well known moments of the beta distribution (Equation 14, where symbol $f(t) \sim g(t)$ indicates the existence of an increasing function h such as $f = h \circ g$).

$$\mathbb{E}\left[q_p(S(t,m))\right] \sim -\sum_{l=0}^{m-1} \mathbb{E}\left[B^p_{i_l(t)/n}\right] \sim \underbrace{-\sum_{l=0}^{m-1}\sum_{j=1}^{n} P(i_l(t)=j)\prod_{k=1}^{p} j+k-1}_{\stackrel{\text{def}}{=} E_p\left(\{i_l(t)\}_{l=0\ldots m-1}\right)} \tag{14}$$

$$\mathscr{P}_2: \quad t \le t' \;\Rightarrow\; E_p\left(\{i_l(t)\}_{l=0\ldots m-1}\right) \le E_p\left(\{i_l(t')\}_{l=0\ldots m-1}\right)$$

The best value for p depends on the quality of the sorting. If the sorting is supposed perfect (the inlier data are top ranked), the best strategy is to minimize the maximum rank in the sample. Thus, the *maximum norm* must be chosen ($p = \infty$). But in the case of a poor sampling criterion, a sample containing many low rank values must be considered better than an other with many high ranks, even if its maximum rank is greater. So, a lower value of p can be used. Our experience shows that $p = 3$ is often close to the optimum. Note that for $(n, p) = (\infty, \infty)$, the sampling procedure becomes the same as PROSAC one. This is feasible in practice but $n = \infty$ would require a complete sorting of the data at each selection.

Now we have a convenient reformulation of $\mathscr{P}_1$ and $\mathscr{P}_2$ (Equation 12 and 14), we can define the function $i_l(t)$ we use. Let $u_1(t), \ldots, u_m(t)$ be the m digits of $\lfloor \frac{t}{T_N} n^m \rfloor$ in base n. $i_l(t)$ defined as follows:

$$i_l(t) = u_l(f(t)) + 1, \quad \forall l \in \{0, \ldots, m-1\} \tag{15}$$

clearly satisfies properties (a) and (b) of Equation 12, for any permutation f of $\{1, \ldots, T_N\}$. The permutation f must be chosen in a way satisfying $\mathscr{P}_2$. f can be precomputed using any sorting algorithm or computed online by means of the fast marching method as described in Figure 4.

Choice of *n*. A key point is that n has not to be set as a function of the number of data points N. It's optimal value depends on the problem complexity (mainly the inlier ratio and the number of different model parametrizations present in the data).

Let $\sigma_{B_{i/n}}$ be the standard deviation of $B_{i/n}$ (Equation 16). The value of $\sigma_{B_{i/n}}$ determines the accuracy of the selection localization in the ranking. It depends directly on n. A complex problem requires an accurate selection as only a thin range of (hopefully) top ranked data are inliers.

$$\sigma_{B_{i/n}} = \frac{1}{n+1}\sqrt{\frac{i(n-i+1)}{n+2}} \tag{16}$$

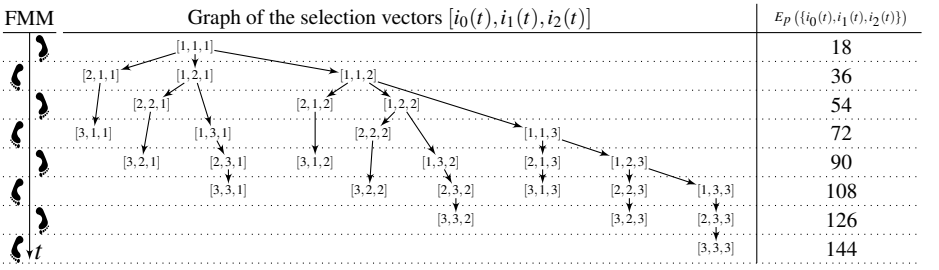


Figure 4: Ordering of the selection vectors $[i_0(t), \ldots, i_{m-1}(t)]$ as a function of $E_p(\{i_0(t), \ldots, i_{m-1}(t)\})$ using the fast marching method (FMM), for $m = 3$, $n = 3$ and $p = 3$.

For any fixed i, $\sigma_{B_{i/n}}$ is asymptotically equivalent to $\frac{\sqrt{i}}{n}$ as n increases. Thus, n must be set proportionally to the inverse of the ratio of data points that would complete properly a sample. For most typical computer vision problems, 10 is a sufficient value of n.

5 Results

5.1 Results on Simulated Data

In Section 3 we present many prior information on the potential of a sample. Here, we evaluate the ability of our sampling algorithm to take this information into account.

We build three theoretical ranking results (as those presented Figure 2). The first is perfect (the data best completing the sample are top ranked). The second is better than random but imperfect (a suitable data point is more likely to be top ranked). The third is purely random (suitable data points are randomly scattered in the ranking). For each one of these ranking qualities, we measure the probability of drawing a suitable sample with $S(t,m)$ during the guided iterations ($t \in \{1, \ldots, T_N\}$). Figure 5 shows that a ranking better than random leads to a high ratio of suitable samples in the first iterations of BetaSAC. In case of random ranking, BetaSAC sampler behaves exactly as the standard RANSAC one.

5.2 Results on Real Data

In this section we demonstrate the potential of our proposed conditional sampling on the problem of homography estimation between two images (sample size m is 4). As we evaluate the sampling only, we use pairs of images with known ground truth and we stop when a model matching with 95% of the inliers is found. Each hypothesis is verified against all data.

The use of BetaSAC requires the definition of a scoring function q. We used two different functions. The first, $q_{matching}(d)$, is simply the matching score of the correspondence d. This is the function used in PROSAC. Let P_d, P'_d and A_d be two end points and the associated affine matrix of a correspondence d obtained with an affine invariant key point detector (we used [8]). Given a partial hypothesis set $s = \{s_1, \ldots, s_{|s|}\}$ and for any correspondence d, the second scoring function, $q_{affine}(d,s)$, is defined in Equation 17. As it depends on s, it could not be used in the PROSAC framework. Figure 2 shows results of correspondence rankings

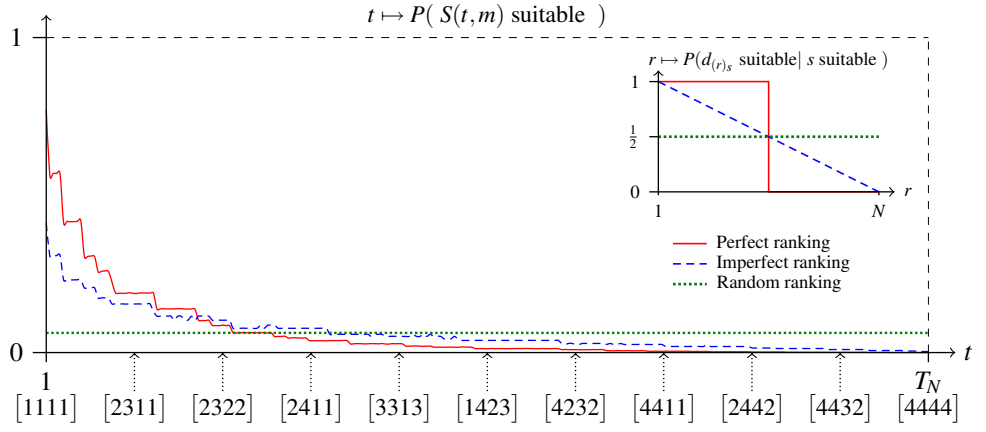


Figure 5: Probability of generating a suitable sample during the iterations of BetaSAC with rankings of different qualities (perfect, imperfect and random). BetaSAC setting is $n = 4$ and $p = 3$. The minimal hypothesis set size m is set to 4 and the inlier ratio is 50%. Selection vectors $[i_0(t), \ldots, i_{m-1}(t)]$ are indicated for some values of t.

obtained with q_{affine}.

$$q_{affine}(d,s) = \begin{cases} q_{matching}(d) & \text{if } |s| = 0 \\ -\left(\|P'_{s_1} - A_d P_{s_1}\| + \|P'_d - A_{s_1} P_d\|\right) & \text{otherwise} \end{cases} \tag{17}$$

Results on four image pairs are presented in Table 5.2. BetaSAC with scoring function q_{affine} is always significantly faster than RANSAC and PROSAC, which prove the benefit of a conditional sampling.

6 Conclusion

This article proposes a unifying notation for the reordered RANSAC sampling methods and introduces a new one, BetaSAC. These methods achieve a guidance of sample drawings while preventing from impairing RANSAC. BetaSAC offers a selection conditional on the previous data selected for a hypothesis set.

We propose to distinguish between four inlier samples types differing in terms of potential to lead to a good model parameters estimation. We discusse how BetaSAC gives the opportunity to generate samples with best potential.

Our algorithm is presented as a general framework in which any kind of information and criteria can be used easily and with a negligible additional computational cost. The only thing needed is a function able to decide between two candidate data points to complement, at best, a forming hypothesis sample.

We demonstrated the benefits of the method on the homography estimation problem. It appears that BetaSAC is equivalent to PROSAC when using the same correspondence inlier prior. With the use of affine matrices associated to the correspondences, BetaSAC is measured to be dozens of times faster.

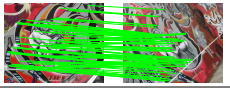
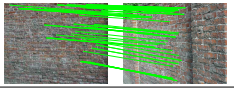
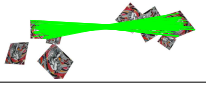
		Sampling algorithm	Mean iters.	Min iters.	Max iters.	Time (ms)	Speed-up
1		RANSAC	8181.9	39	40910	1595.5	1
		PROSAC	1709.9	31	7628	333.3	4.79
		BetaSAC with $q_{matching}$	1548.6	24	8142	289.3	5.51
		BetaSAC with q_{affine}	287.0	3	1840	55.51	**28.74**
2		RANSAC	15858.5	338	90974	1447.4	1
		PROSAC	6445.6	30	39564	587.5	2.46
		BetaSAC with $q_{matching}$	6414.0	32	41183	602.3	2.4
		BetaSAC with q_{affine}	636.9	1	6697	63.8	**22.69**
3		RANSAC	1302.8	9	6179	1293.0	1
		PROSAC	1533.6	129	3990	1523.1	0.85
		BetaSAC with $q_{matching}$	678.2	4	2601	677.9	1.91
		BetaSAC with q_{affine}	30.8	1	157	30.9	**41.85**
4		RANSAC	687.2	1	4975	198.0	1
		PROSAC	329.7	34	1156	94.9	2.09
		BetaSAC with $q_{matching}$	242.8	2	976	71.0	2.79
		BetaSAC with q_{affine}	7.1	1	29	2.12	**93.41**

Table 1: Homography estimation ($m = 4$). All the results are averaged over 100 runs. BetaSAC parametrization is $n = 10$ and $p = 3$. The number of guided iterations T_N is set to 200000. Images pairs 1 and 2 are strong homographies. Pairs 3 and 4 are semi-synthetic examples showing the benefit of our sampling in presence of repeated patterns and different coexisting model parametrizations.

References

[1] T. Botterill, S. Mills, and R. Green. New conditional sampling strategies for speeded-up ransac. In *BMVC09*, 2009.

[2] Ondrej Chum and Jiri Matas. Matching with prosac " progressive sample consensus. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*, pages 220–226, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2372-2. doi: http://dx.doi.org/10.1109/CVPR.2005.221.

[3] Ondrej Chum, Tomas Werner, and Jiri Matas. Two-view geometry estimation unaffected by a dominant plane. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*, pages 772–779, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2372-2. doi: http://dx.doi.org/10.1109/CVPR.2005.354.

[4] 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.

[5] Jan-Michael Frahm and Marc Pollefeys. Ransac for (quasi-)degenerate data (qdegsac). In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 453–460, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2597-0. doi: http://dx.doi.org/10.1109/CVPR.2006.235.

[6] Stephane Laveau and Olivier Faugeras. Oriented projective geometry for computer vision. In *ECCV96*, pages 147–156. Springer-Verlag, 1996.

[7] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004. ISSN 0920-5691. doi: http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94.

[8] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *Int. J. Comput. Vision*, 60(1):63–86, 2004. ISSN 0920-5691. doi: http://dx.doi.org/10.1023/B:VISI.0000027790.02288.f2.

[9] 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 *BMVC02*, page Computer Vision Tools, 2002.

[10] Kai Ni, Hailin Jin, and Frank Dellaert. Groupsac: Efficient consensus in the presence of groupings. In *ICCV09*, Kyoto;Japan, October 2009. URL http://frank.dellaert.com/pubs/Ni09iccv.pdf.

[11] Tomas Pajdla, Tomas Werner, and Vaclav Hlavac. Oriented projective reconstruction, 1998.

[12] Ben J. Tordoff and David W. Murray. Guided-mlesac: Faster image transform estimation by using matching priors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(10):1523–1535, 2005. ISSN 0162-8828. doi: http://dx.doi.org/10.1109/TPAMI.2005.199.

[13] Tomas Werner and Tomas Pajdla. Oriented matching constraints, 2001.

[14] Lihi Zelnik-Manor and Michal Irani. Multiview constraints on homographies. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(2):214–223, 2002. ISSN 0162-8828. doi: http://dx.doi.org/10.1109/34.982901.