

RODIAN: Robustified Median

Seong Hun Lee and Javier Civera

Abstract—We propose a robust method for averaging numbers contaminated by a large proportion of outliers. Our method, dubbed RODIAN, is inspired by the key idea of MINPRAN [1]: We assume that the outliers are uniformly distributed within the range of the data and we search for the region that is least likely to contain outliers only. The median of the data within this region is then taken as RODIAN. Our approach can accurately estimate the true mean of data with more than 50% outliers and runs in time $O(n \log n)$. Unlike other robust techniques, it is completely deterministic and does not rely on a known inlier error bound. Our extensive evaluation shows that RODIAN is much more robust than the median and the least-median-of-squares. This result also holds in the case of non-uniform outlier distributions.

Index Terms—Averaging, median, measure of central tendency, robust statistics, outlier-resistant method.

I. INTRODUCTION

AVERAGING means finding the most representative value of a given set of data points. For one-dimensional numbers, it can be the arithmetic mean, the median or other measures of central tendency. All these measures have different properties, one of which is the robustness to outliers. This is an important property to consider because outliers can severely degrade the averaging accuracy if not handled properly. Robust averaging is a useful technique in a wide variety of domains, including pattern recognition [2], [3], [4], image processing [5], [6], [7], 3D computer vision [8], [9], [10], biomedical engineering [11], [12], [13], economics/econometrics [14], [15], [16], information science [17], [18], [19], environmental studies [20], [21], geochemistry [22], [23], forensic science [24], psychology [25] and database research [26].

If the given data set contains a small number of outliers, it may be sufficient to use the median, as, contrary to the mean, it is robust up to a certain outlier ratio [27]. The median can be considered as a specific case of the alpha-trimmed mean with $\alpha = 0.5$. The alpha-trimmed mean [28] of n numbers is defined as the arithmetic mean after truncating the largest $\frac{\alpha}{2}n$ and the smallest $\frac{\alpha}{2}n$ elements. This method assumes that the outliers are likely to be located at the high and low ends of the sorted data. As a result, it fails when a large number of outliers are located mostly on one side of the long tails.

To handle such cases, a more elaborate method should be used. One popular example is the maximum likelihood-type estimator (M-estimator) [29]. Paired with iteratively reweighted least squares (IRLS) [30], it can effectively down-weight the influence of outliers. However, M-estimators, such as the Huber function, often require a control parameter to

be carefully tuned to the inlier error distribution. Also, their robustness strongly depends on the initial seed, and accurate initialization in the presence of many outliers is already a non-trivial problem in and of itself.

Another popular robust estimation method is RANSAC [31]. It involves random sampling, but it can be made deterministic for the 1D averaging problem if we simply pick every number as a sample once. While this method can handle a very large number of outliers, it incurs a computational cost of $O(n^2)$ and requires the prior knowledge of the inlier error bound.

The least-median-of-squares (LMedS) [32], on the other hand, does not require any prior knowledge. For the 1D averaging problem, the LMedS can be obtained by finding the data point that yields the smallest median deviation from the rest. This would involve $O(n^2 \log n)$ computations. Like the median, the LMedS has a breakdown point of 50%.

Another method that does not rely on a known inlier error bound is MINPRAN [1]. This method is more robust than the LMedS, as it can handle more than 50% outliers. However, it is slower than the LMedS and has a random nature.

In this work, we propose RODIAN, a novel robust measure of central tendency. Our method is inspired by the core idea of MINPRAN [1]: We assume that the outliers follow a uniform distribution and find the median in the bounded region that is least likely to contain outliers only. Unlike MINPRAN, however, our method is deterministic and runs in time $O(n \log n)$.¹ Also, unlike RANSAC [31] and Huber-like cost functions [29], no parameter tuning is needed to account for different inlier distributions. Our experiments show that RODIAN can handle more than 50% outliers, outperforming the median and the LMedS [32] in terms of robustness. We release our code at <https://seonghun-lee.github.io>.

The remainder of this paper is organized as follows: We detail our method in Section II and present the evaluation results in Section III. In Section IV, we discuss the limitation of our work. Finally, conclusions are drawn in Section V.

II. METHOD

A. Main Idea

Suppose that we are given a set of numbers. Each number is either an inlier or an outlier, but we do not know which is which. Assuming that the inliers are scattered around a certain number μ , how can we estimate μ from this noisy, outlier-contaminated data? Our approach is to find the most densely populated region in the data and take the median value in that region. Now the question is how to determine this region.

¹This is made possible due to the following differences from [1]: First, we do not use random sampling. Second, we do not compute any residuals. Third, we evaluate the probability associated with k inliers, rather than *at least* k inliers. Interested readers are referred to [1] for a closer examination of how MINPRAN is different from our method described in Section II.

This work was supported in part by the Spanish government (project PGC2018-09637-B-I00) and in part by the Aragon regional government (DGA_FSE T45_20R).

The authors are with the University of Zaragoza, Zaragoza 50018, Spain (e-mail: seonghunlee@unizar.es; jcivera@unizar.es).

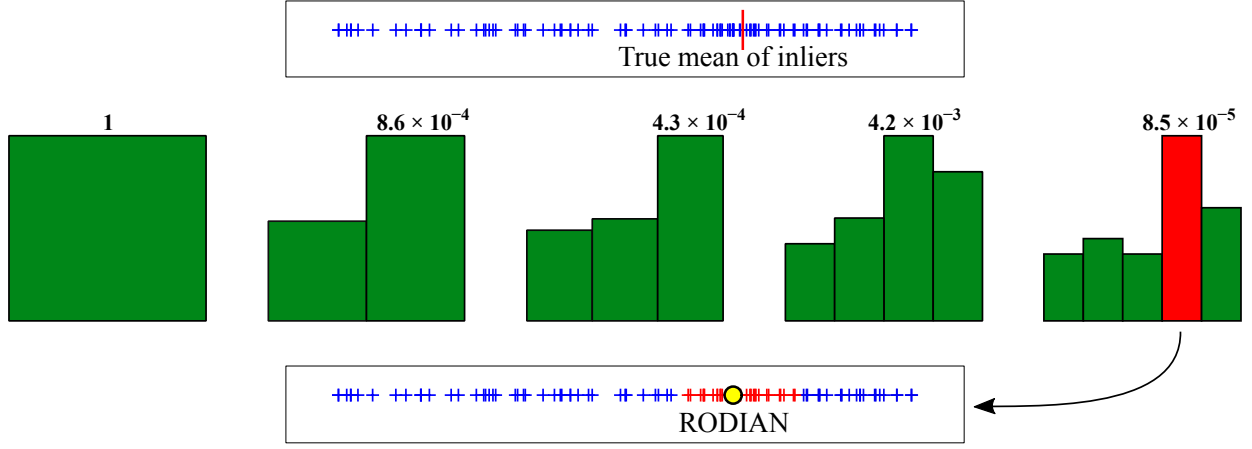


Fig. 1. **Top:** 100 numbers used as input for a toy example. 80% of the numbers are outliers, uniformly distributed between 0 and 100. The inliers follow $\mathcal{N}(70, 5^2)$. **Middle:** We build multiple histograms with a varying number of bins (1–5 in this example). For each histogram, we find the highest bin and evaluate the probability of this occurring purely by chance, *assuming that the outliers are uniformly distributed*. These probabilities of randomness are given above each histogram. We find the bin that produces the smallest probability (shown in red). **Bottom:** RODIAN is the median of the numbers in this bin.

One simple heuristic approach is to build a histogram and find the tallest bin. Then, the edges of this bin correspond to the upper and lower bounds of the densest region. This is a reasonable approach, but there is one problem: The histogram can be constructed in many different ways. If we constrain the lower edge of the first bin to be the minimum value and the upper edge of the last bin to be the maximum value, then we can obtain multiple histograms by varying the number of bins. So, which histogram is the right one to use?

Our answer to this question is that we choose the histogram with the bin that is least likely to have occurred by chance. For any histogram, each of its bins have its associated probability of randomness. For example, if all bins have the exact same height, we can deduce that the numbers are uniformly distributed, and thus random in this sense. By the same token, if one of the bins is significantly taller than the others, then it is unlikely that it occurred due to the randomness. In other words, this very tall bin has a low probability of randomness.

Essentially, what we propose is to build multiple histograms with the different numbers of bins, evaluate the probability of randomness associated with the tallest bin of each histogram, and choose the one that yields the smallest probability of randomness. This is because the minimum probability of randomness implies the maximum probability of containing mostly inliers. This process is illustrated in Fig. 1.

The remaining question is how exactly we compute this probability of randomness. Basically, we adopt a similar idea proposed by Stewart [1] and compute the probability in a binomial distribution, assuming that the random outliers are uniformly distributed across the entire range. Note that if a trial has a probability of success p , the probability of obtaining k successes from n trials is given by

$$P(k) = \binom{n}{k} p^k (1-p)^{n-k}. \quad (1)$$

In our problem, p is the probability of a random outlier falling inside the tallest bin, k is the frequency of this bin, and n is the size of the data. Since an outlier can fall inside any other

bins with an equal probability, we have $p = 1/b$ where b is the number of bins. Therefore, the probability of randomness associated with the bin containing k numbers is given by

$$P(k) = \binom{n}{k} \left(\frac{1}{b}\right)^k \left(1 - \frac{1}{b}\right)^{n-k}. \quad (2)$$

In summary, we vary b , find k by building a histogram, compute $P(k)$, and repeat this process until we find the value of b that leads to the smallest $P(k)$. In the next section, we discuss several strategies we came up with to improve the efficiency of the algorithm.

B. Implementation Details

1. How many histograms do we consider?:

According to the pigeonhole principle, if we set the number of bins to $n - 1$, at least one bin will contain more than one number. Therefore, one could find the theoretically optimal number of bins, b^* , by varying b from 1 to $n - 1$, searching for the minimum $P(k)$ in Eq. (2). However, in our experiment described in Section III, we found that setting $b > 20$ hardly makes any difference in the final accuracy. We also empirically found that there is no need to try all integers between 1 and 20, as similar results could be obtained faster with $b \in \{2, 3, 4, 5, 7, 9, 11, 14, 17, 20\}$.

2. How to accelerate the histogram building process:

Building multiple histograms one by one can take a long time. For efficiency, we precompute a table that matches the bin edges and the bin indices of all the histograms. This process is explained Fig. 2. In order to reuse this table on any data, it must be agnostic of the input. To this end, we normalize the input data such that its range becomes $[0, 1]$. This way, all edges get predetermined values between 0 and 1.

C. Summary

- 1) Precompute a table, as described in Fig. 2, with $A = 0$, $B = 0.5$, $C = 1$, etc. For the number of bins, we use $b \in \{2, 3, 4, 5, 7, 9, 11, 14, 17, 20\}$.
- 2) Sort and normalize the input such that its range is between 0 and 1, *i.e.*,

$$x_i \leftarrow \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \quad \text{for } i = 1, 2, \dots, n. \quad (3)$$

- 3) For each data point, use the precomputed table to find the corresponding bin index in each histogram.
- 4) For each histogram, find the frequency of the tallest bin and the associated probability of randomness (Eq. (2)).
- 5) Find the histogram that leads to the smallest probability.
- 6) In that histogram, find the median of the numbers that fall inside the tallest bin.
- 7) Unnormalize this median. The final value is RODIAN.

In Step 4, we discard a histogram if multiple bins have the same maximum frequency. If, by any chance, all histograms are discarded, we simply take the median of the original input.

Time analysis: The time complexity of Step 2 and 6 is $O(n \log n)$. The other steps run in either $O(1)$ or $O(n)$. Hence, the total time complexity is $O(n \log n)$.

III. RESULTS

We compare RODIAN with three other methods:

- 1) Median,
- 2) Least-median-of-squares (LMedS) [32], estimated as the data point with the smallest median (squared) distance to the rest, *i.e.*,

$$\text{LMedS} = \arg \min_{x_i} \text{med}_j (x_i - x_j)^2, \quad (4)$$

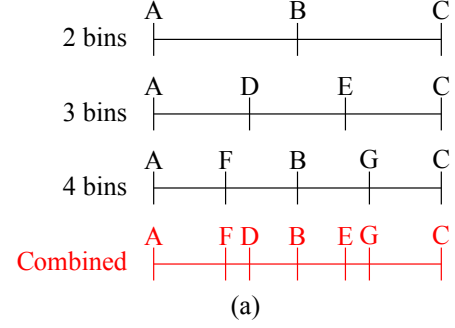
- 3) Median of the tallest bin of a fixed histogram, obtained in the following way: (i) Build a histogram with a fixed number of bins, (ii) Collect all the numbers that fall inside the tallest bin, (iii) Compute their median.

We present the results on synthetic data with two different outlier distributions: a uniform distribution (Fig. 3) and a mixture of a uniform and a Gaussian distribution (Fig. 4). In both cases, RODIAN outperforms the rest in terms of robustness. Especially, Fig. 4 shows that even though we assumed a uniform outlier distribution in our derivation of RODIAN, it can still handle non-uniform outliers relatively well if σ_{outlier} is larger than σ_{inlier} .

Table I compares the accuracy of RODIAN and the fixed-histogram approach for low to moderate outlier ratios. It again demonstrates the advantage of using RODIAN over a fixed histogram. In Fig. 5, we plot the mean computation times of the median, LMedS and RODIAN. We observe that RODIAN is much more scalable than the LMedS.

IV. LIMITATION

The main limitation of RODIAN is that its accuracy slightly drops when there are too few outliers (see Table I). This happens because the densest region of the inlier distribution is not always well aligned with the location of the true mean.



(b)

	A-F	F-D	D-B	B-E	E-G	G-C
# bins 2	1	1	1	2	2	2
3	1	1	2	2	3	3
4	1	2	2	3	3	4

Fig. 2. This example illustrates the process of precomputing the tabular data for building histograms: (a) Suppose that we want to build three histograms with 2, 3 and 4 bins. We collect the bin edges of all histograms (without duplicates) and sort them in a single array, *i.e.*, $[A, F, D, B, E, G, C]$. (b) We assign the corresponding bin index to each region bounded by two successive edges in the combined array. For example, a number between edge F and D would fall inside the 1st bin in the 2-bin and 3-bin histogram, and 2nd bin in the 4-bin histogram. These indices are given in the F-D column. By checking which region a number falls in, we can immediately find the corresponding bin index in each histogram.

While this is certainly not a favorable property, the average error increase is relatively small (around 10% of the standard deviation of the inliers in Table I). We believe that this is a tolerable level in outlier-prone situations, which is the main domain we target in this work.

One potential solution is to detect when the data is outlier-free and switch to a traditional median. If the type of the inlier distribution is known (*e.g.*, Gaussian), one can use a statistical test to check if the whole data follow the inlier distribution (*e.g.*, normality test [33], [34]). In this work, however, we aim to make our method generalizable to any types of inlier distribution as long as it is unimodal. Discerning outlier-free data in such a general scenario is left for future work.

V. CONCLUSION

In this work, we presented RODIAN, a novel method for averaging outlier-contaminated numbers. It consists of two main steps: (1) determine a bounded region in the range that would contain mostly inliers, and (2) find the median within that region. The key idea of the first step is to assume a uniform outlier distribution and search for the region that is least likely to have occurred due to outliers. Unlike MINPRAN [1], where a similar idea was used, our method is deterministic and runs in time $O(n \log n)$. Unlike RANSAC [31] and Huber-like loss functions [29], we do not need to tune a control parameter to adapt to different inlier error distributions. Finally, unlike the median and the LMedS [32], RODIAN can handle more than 50% outliers. An extensive evaluation demonstrates its excellent robustness, versatility and scalability.

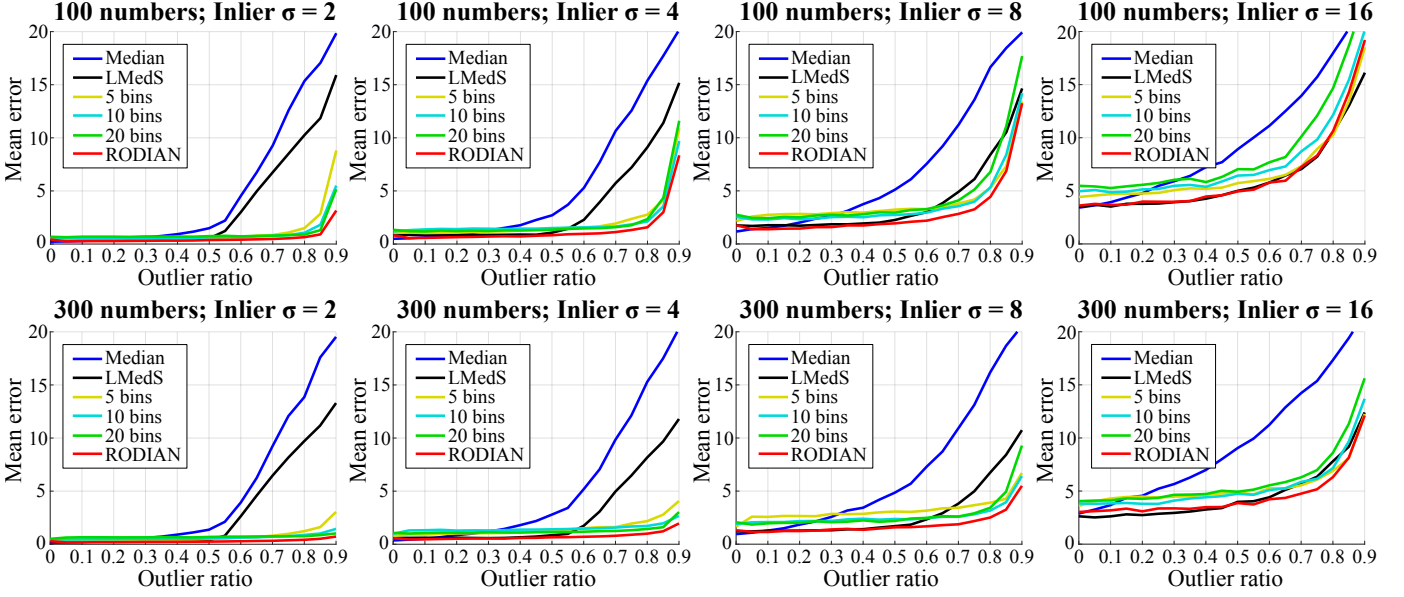


Fig. 3. Mean error comparison under a **uniform outlier distribution**: The evaluation is carried out with 100 numbers (top row) and 300 numbers (bottom row). The numbers are generated such that the inliers follow $\mathcal{N}(\mu, \sigma^2)$ where $0 < \mu < 100$ and $\sigma \in \{2, 4, 8, 16\}$, and the outliers follow $\mathcal{U}(0, 100)$. If any number is outside a range $[0, 100]$, it is removed and regenerated until all numbers are within this range. Each data point in the graph represents the mean error of 1000 independent runs. It shows that, across different inlier noise levels, RODIAN is the most robust to outliers. When the inlier noise is small, the breakdown point of RODIAN is well over 80% (see the first column). Also, RODIAN is generally more accurate than the fixed-histogram approach.

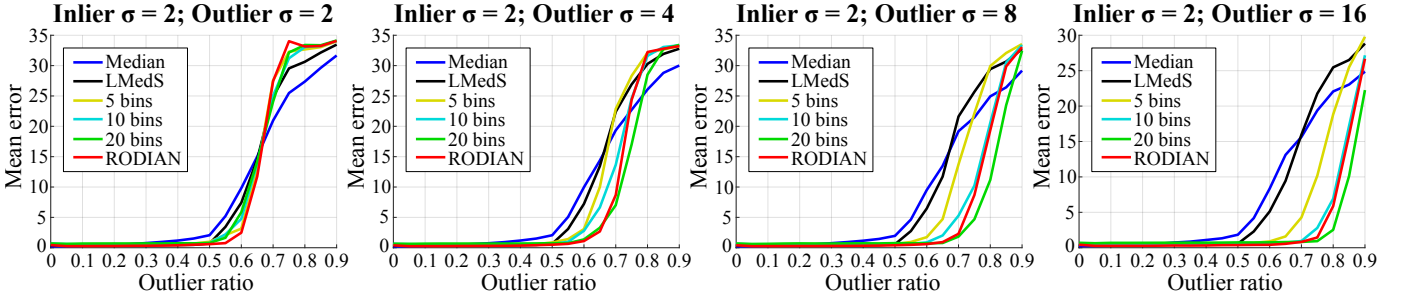


Fig. 4. Mean error comparison under a **uniform + Gaussian outlier distribution**: The evaluation is carried out with 100 numbers. The numbers are generated using the same procedure described in Fig. 3, except that half of the outliers now follow $\mathcal{N}(\mu_{\text{outlier}}, \sigma_{\text{outlier}}^2)$, $0 < \mu_{\text{outlier}} < 100$. We observe that RODIAN has a higher breakdown point than the median and the LMedS. When the outlier ratio is very high, the fixed-histogram approach with 20 bins produces smaller errors than RODIAN. However, they both are well beyond the breakdown point there and comparisons are meaningless, as errors are driven by outliers.

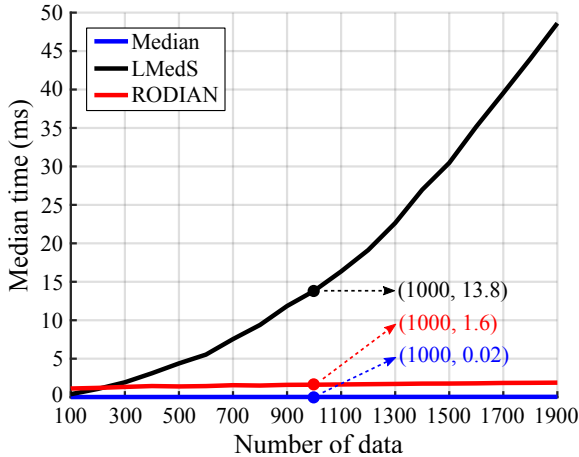


Fig. 5. Median computation times (ms) of 1000 runs: The time complexity of the median, LMedS and RODIAN is $O(n \log n)$, $O(n^2 \log n)$ and $O(n \log n)$, respectively. The median is always the fastest, and RODIAN is faster than the LMedS when $\#data \gtrsim 200$. All methods are implemented in MATLAB and run on a laptop with an Intel's 4th Gen i7 CPU (2.8 GHz).

TABLE I
USING FIXED HISTOGRAMS vs. RODIAN

Outliers		5 bins	10 bins	20 bins	30 bins	50 bins	RODIAN
Uniform ^a	0%	0.54	0.64	0.69	0.74	0.80	0.42
	10%	0.36	0.50	0.69	0.61	0.61	0.26
	20%	0.39	0.51	0.69	0.63	0.62	0.28
	30%	0.43	0.54	0.70	0.64	0.63	0.30
	40%	0.48	0.57	0.71	0.67	0.66	0.32
	50%	0.56	0.61	0.72	0.68	0.68	0.36
Gaussian ^b	0%	0.54	0.63	0.69	0.73	0.79	0.42
	10%	0.46	0.64	0.63	0.64	0.68	0.28
	20%	0.47	0.64	0.65	0.66	0.69	0.30
	30%	0.47	0.65	0.68	0.67	0.72	0.33
	40%	0.50	0.71	0.77	0.77	0.85	0.38
	50%	4.36	2.33	2.29	1.93	2.49	1.34

We generate 100 numbers within a range $[0, 100]$ and average them using either a fixed histogram or RODIAN. This is repeated 10000 times. RODIAN produces the smallest mean error.

^aInliers follow $\mathcal{N}(\mu, 2^2)$ with $0 < \mu < 100$.

^bInliers and outliers follow $\mathcal{N}(\mu_1, 2^2)$ and $\mathcal{N}(\mu_2, 4^2)$ with $0 < \mu_1, \mu_2 < 100$. (Note: This dataset is different from that of Fig. 4.)

REFERENCES

- [1] C. Stewart, "MINPRAN: a new robust estimator for computer vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 10, pp. 925–938, 1995. 1, 2, 3
- [2] S. Hauberg, A. Feragen, and M. J. Black, "Grassmann averages for scalable robust pca," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3810–3817. 1
- [3] X. Li, S. Fei, and T. Zhang, "Median MSD-based method for face recognition," *Neurocomputing*, vol. 72, no. 16, pp. 3930–3934, 2009. 1
- [4] T. Lewis, R. Owens, and A. Baddeley, "Averaging feature maps," *Pattern Recognition*, vol. 32, no. 9, pp. 1615–1630, 1999. 1
- [5] V. Vaish, M. Levoy, R. Szeliski, C. Zitnick, and S. B. Kang, "Reconstructing occluded surfaces using synthetic apertures: Stereo, focus and robust measures," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2006, pp. 2331–2338. 1
- [6] K. N. Chaudhury and A. Singer, "Non-local euclidean medians," *IEEE Signal Processing Letters*, vol. 19, no. 11, pp. 745–748, 2012. 1
- [7] S. H. Khatoonabadi and I. V. Bajic, "Video object tracking in the compressed domain using spatio-temporal Markov random fields," *IEEE Transactions on Image Processing*, vol. 22, no. 1, pp. 300–313, 2013. 1
- [8] M. Gesto Diaz, F. Tombari, P. Rodriguez-Gonzalez, and D. Gonzalez-Aguilera, "Analysis and evaluation between the first and the second generation of rgb-d sensors," *IEEE Sensors Journal*, vol. 15, no. 11, pp. 6507–6516, 2015. 1
- [9] S. H. Lee and J. Civera, "Robust single rotation averaging," *CoRR*, vol. abs/2004.00732, 2020. 1
- [10] Z. Cui and P. Tan, "Global structure-from-motion by similarity averaging," in *IEEE Int. Conf. Comput. Vis.*, 2015, pp. 864–872. 1
- [11] Z. Leonowicz, J. Karvanen, and S. L. Shishkin, "Trimmed estimators for robust averaging of event-related potentials," *Journal of neuroscience methods*, vol. 142, no. 1, pp. 17–26, 2005. 1
- [12] J. M. Leski, "Robust weighted averaging [of biomedical signals]," *IEEE Trans. Biomed. Eng.*, vol. 49, no. 8, pp. 796–804, 2002. 1
- [13] K. Kotowski, K. Stapor, and J. Leski, "Improved robust weighted averaging for event-related potentials in EEG," *Biocybernetics and Biomedical Engineering*, vol. 39, no. 4, pp. 1036–1046, 2019. 1
- [14] M. F. Bryan, S. G. Cecchetti, and R. L. Wiggins, "Efficient inflation estimation," National Bureau of Economic Research, Tech. Rep. 6183, 1997. 1
- [15] P. A. Mykland and L. Zhang, "Between data cleaning and inference: Pre-averaging and robust estimators of the efficient price," *Journal of Econometrics*, vol. 194, no. 2, pp. 242–262, 2016. 1
- [16] J. Dias Curto, "Averages: There is still something to learn," *Computational Economics*, 2021. 1
- [17] P. Angelov and R. Yager, "Density-based averaging – a new operator for data fusion," *Information Sciences*, vol. 222, pp. 163–174, 2013, including Special Section on New Trends in Ambient Intelligence and Bio-inspired Systems. 1
- [18] F. Garcin, B. Faltings, and R. Jurca, "Aggregating reputation feedback," *Proceedings of the First International Conference on Reputation: Theory and Technology*, 2009. 1
- [19] G. Beliakov, H. B. Sola, and T. Calvo, *A Practical Guide to Averaging Functions*, ser. Studies in Fuzziness and Soft Computing. Springer, 2016, vol. 329. 1
- [20] C. Zhang and S. Zhang, "A robust-symmetric mean: A new way of mean calculation for environmental data," *GeoJournal*, vol. 40, pp. 209–212, 1996. 1
- [21] N. Merchant, A. Farcas, and C. Powell, "Acoustic metric specification," Centre for Environment, Fisheries & Aquaculture Science (Cefas), UK, Tech. Rep., 2018. 1
- [22] N. Rock, J. Webb, N. McNaughton, and G. Bell, "Nonparametric estimation of averages and errors for small data-sets in isotope geoscience: a proposal," *Chemical Geology: Isotope Geoscience section*, vol. 66, no. 1, pp. 163–177, 1987. 1
- [23] N. M. S. Rock, "Summary statistics in geochemistry: A study of the performance of robust estimates," *Mathematical Geology*, vol. 20, no. 3, pp. 243–275, 1988. 1
- [24] M. Illes and M. Boué, "Robust estimation for area of origin in bloodstain pattern analysis via directional analysis," *Forensic Science International*, vol. 226, no. 1, pp. 223–229, 2013. 1
- [25] D. S. Courvoisier and O. Renaud, "Robust analysis of the central tendency, simple and multiple regression and ANOVA: a step by step tutorial," *International Journal of Psychological Research*, vol. 3, no. 1, p. 78–87, Jun. 2010. 1
- [26] J. M. Hellerstein, "Quantitative data cleaning for large databases," 2008. 1
- [27] M. A. Davis and D. L. P. Jr., "Central tendencies, measures of," International Encyclopedia of the Social Sciences dictionary, 2022. [Online]. Available: <https://www.encyclopedia.com/social-sciences/applied-and-social-sciences-magazines/central-tendencies-measures> 1
- [28] J. Bednar and T. Watt, "Alpha-trimmed means and their relationship to median filters," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 1, pp. 145–153, 1984. 1
- [29] P. J. Huber, *Robust Statistics*, ser. Wiley Series in Probability and Statistics. Wiley, 1981. 1, 3
- [30] P. W. Holland and R. E. Welsch, "Robust regression using iteratively reweighted least-squares," *Communications in Statistics - Theory and Methods*, vol. 6, no. 9, pp. 813–827, 1977. 1
- [31] M. A. Fischler and R. C. 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. 1, 3
- [32] P. J. Rousseeuw, "Least median of squares regression," *Journal of the American Statistical Association*, vol. 79, no. 388, pp. 871–880, 1984. 1, 3
- [33] H. Thode, *Testing For Normality*, ser. Statistics, textbooks and monographs. CRC Press, 2002. 3
- [34] S. S. Shapiro and M. B. Wilk, "An analysis of variance test for normality (complete samples)," *Biometrika*, vol. 52, no. 3/4, pp. 591–611, 1965. 3