IBIMA Publishing
Communications of International Proceedings
https://ibimapublishing.com/p-articles/45AI/2025/4540125/
Vol. 2025 (20), Article ID 4540125

Visual Analysis of Dispersion Of Multidimensional Scattered Data*

Janusz OPILA

AGH University of Krakow, Krakow, Poland

Correspondence should be addressed to: Janusz OPILA, jmo@agh.edu.pl

* Presented at the 45th IBIMA International Conference, 25-26 June 2025, Cordoba, Spain

Abstract

One of the goals of data dispersion analysis is the determination of the empirical probability density function based on the sampled data. The paper presents a critical discussion of four methods of analysis of the dispersion of multivariate data in the sample: classical histogram, ”inverted histogram”, ”smoothed histogram” based on the kernel density estimation function (KDE) with the Gaussian kernel, and the newly developed and hereby proposed KDE with ”algorithmic kernel” based on local density approach (aKDE). The purpose of the work is to inspect the strengths and weaknesses of the methods under consideration. In particular, the hypothesis is verified that the proposed method, aKDE, is an acceptable alternative to more classic methods. The considerations are illustrated by calculations and graphs made on samples drawn from the mixed multivariate correlated distribution for one, two, and three dimensions.

Keywords: KDE, algorithmic kernel, data visualization, data dispersion

Introduction

Data visualization is an important part of data analysis in numerous areas, such as science, business, econometrics, and teaching Várkoly (2017). The applications of visual data analysis, including data dispersion analysis, are varied. In particular, it can be useful for the initial visual determination of possible components of a mixed probability distribution, making it a step to identify the number and location of possible data clusters. Another important application is the determination of the empirical Probability Density Function (PDF), crucial for Monte Carlo simulation methods (Opiła (2019), Opiła (2020)). The most widely used method is the classical histogram method despite its numerous weaknesses. In particular, at least five counts per histogram bin are required. For one-dimensional data, this requires at least 50 data points on 10 bins design, but for two-dimensional data it is at least 500 measurements (10×10 grid), and for 3D grids the number grows to 5000 data points (10×10×10 grid). Unfortunately, in many disciplines (sciences of quality and management, biology, economics), such a number is unattainable. Therefore, it is important to search for algorithms that can give an acceptable PDF estimate with substantially fewer data or a better estimation using the same data.

Another problem with a classical histogram is finding the proper number and size of the bins. Although KDE overcomes this difficulty, it introduces another one, namely the choice of the kernel function and choice of the value of the ”bandwidth”. In fact there are several kernels available, eg. gaussian, cosine, box, fermi[1)] but still there is no clear hint which one is best for the given data or how to select a suitable bandwidth. This motivated me to search for a novel method that would overcome the difficulties mentioned above.

The aim of the work is the critical analysis of the strengths and weaknesses of the current and new solution by applying the method to simulated data sets. In addition, the hypothesis is verified that the proposed method, namely aKDE, is an acceptable alternative to classic methods. Thus, the paper critically compares four methods

Cite this Article as: Janusz OPILA, Vol. 2025 (20) ”Visual Analysis of Dispersion Of Multidimensional Scattered Data" Communications of International Proceedings, Vol. 2025 (20), Article ID 4540125, https://doi.org/10.5171/2025.4540125

of visualization and dispersion analysis of multidimensional data: the classical histogram, the so-called ”inverted histogram”, the ”smoothed” KDE histogram with the Gaussian kernel, and with the newly developed algorithmic kernel, aKDE (algorithmic Kernel Density Estimate). The considerations are illustrated with calculations and graphs made on a random sample drawn from mixed multi-component multivariate correlated probability distributions.

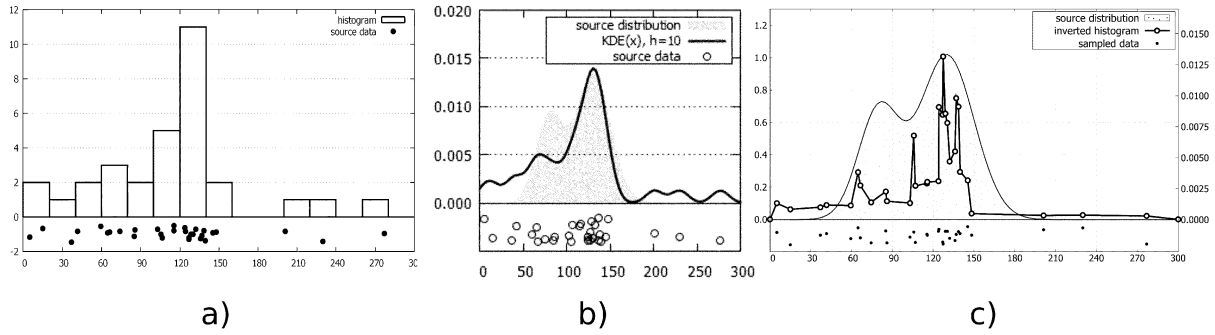


Figure 1: Various techniques for visualizing one-dimensional scattered data. Figure a) - visualization using the classic histogram method (Statistica (2016)), b) smoothed histogram (KDE) (gnuplot (2024)), c) - visualization using the inverse histogram method (gnuplot (2024)). All figures are supplemented with a scatter plot of the same measurement data (below X axis). *Source: own study, based on Opila (2018)*

Visualization of Data Dispersion

Presently researcher has only a few techniques to visualize sparse or randomly distributed data. In the one-dimensional case, one can, as mentioned before, use one of four methods:

1. scatter plot with a random ordinate to avoid overlapping measurement points (Fig. 1, bottom),
2. regular histogram plot (Fig.1a),
3. KDE smoothed histogram plot (Fig.1b),
4. inverted histogram plot, discussed in the following section (Fig.1c).

In the two-dimensional case, the following are available:

1. a histogram of two variables (Fig.2a),

2. a scatter plot of raw data (Fig.2b),

3. various techniques based on the KDE kernel smoothing functions, employing surface plot, contour plot or heat map (Fig.2c),

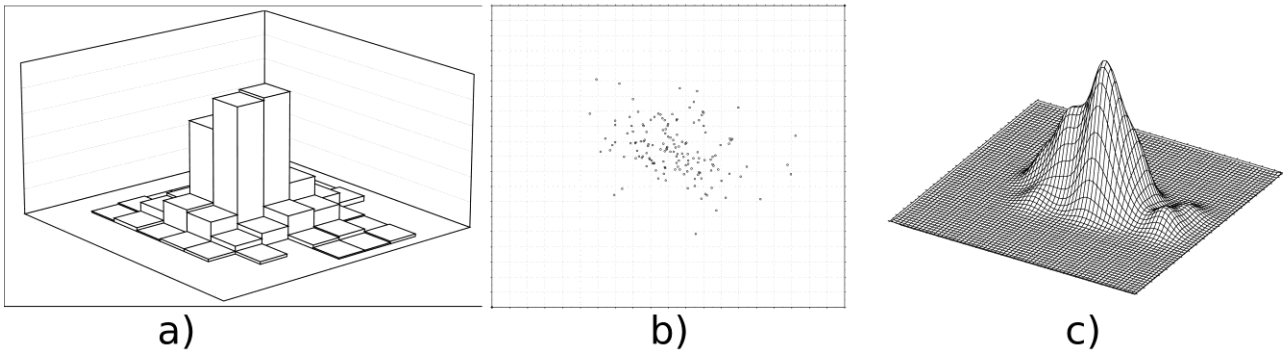


Figure 2: Various techniques for visualizing multidimensional scattered data. Figure a) visualization using the classic histogram method (Statistica (2016)), b) scattered point plot (gnuplot (2024)), c) visualization of the same data using the KDE employing surface plot (gnuplot (2024)). ***Source: own study, based on Opila (2018)***

As illustrated in Fig.3, classic histogram of two variables does not have analytical value. Despite the large size of the data set (up to 1200 points) and the number of class intervals (up to 100), the only conclusion is that the attribute studied has a distribution that is somehow symmetrical and centered around one point, which is not true in the tested case (the sample was drawn from a mixed distribution).

The XY scatter plot also does not contribute much because, on the one hand, subsequent points overlap, making the analysis pointless, and, on the other hand, it is difficult to indicate the central point of the distribution (means: μX, μY).

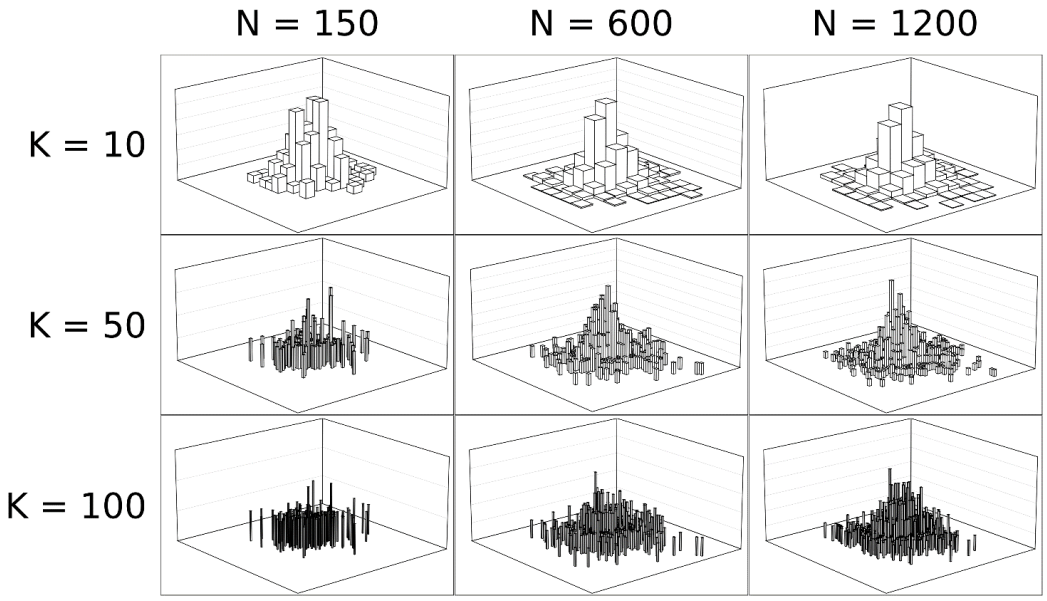


Figure. 3: Comparison of histograms of two variables for different numbers of class intervals (K=10/50/100) and with respect to the number of data in the sample (N=150/600/1200). Increasing the number of intervals does not improve the analytical value of the visualization (Statistica (2016). ***Source: own study***

For the 3D case there are two options:
1. a 3D scatter plot,
2. various techniques based on the KDE kernel smoothing functions;

In 3D case PDF is defined over the 3D manifold, thus for visualization volume rendering techniques are required, eg. isosurface and cross sections (Fig.4). KDE is used as a method to process a set of irregularly distributed measurement points in a regular grid based on the so-called ”kernel function”. However, since it cannot be derived from the first principles, its significance is reduced to a convenient method to illustrate the dispersion of data. The inverse histogram technique cannot be applied in more than one dimension because the required distance cannot be computed (Sect.3). This was a motivation to search for an upgrade of the inverse histogram algorithm to multidimensional spaces. The solution, called algorithmic KDE (aKDE), is discussed in the next section.

Definition of the ”aKDE” algorithm

The definition of an ”inverted histogram” results from the formula Opiła (2018) (1):

$$N_i \cong N_0 \cdot f(x)\,\Delta x_i \tag{1}$$

where $f(x)$ is an estimate of the local probability density, N_i denotes the data count in the interval Δx_i and N_0 is the total sample size. Thus, the local PDF =$f(x)$ can be derived as (2):

$$f(x) \cong \frac{N_i}{N_0} \cdot \frac{1}{\Delta x_i} \tag{2}$$

In the case of a regular histogram, it is assumed that the width of the class interval is constant $\Delta x_i = const$ and that the number N_i is counted. In case of a small sample size, this can cause large errors in the distribution estimation, including false peaks. In addition, there is no acceptable theoretical estimate of the proper width of the class interval, depending on the number of data and their range (see Scott (1979), Sturges (1926), Doane (1976)). Meanwhile, instead of calculating the number N_i in the class intervals, the problem may be reversed by examining the width of the interval necessary to collect a fixed number of data. Rewriting (2), we obtain formula (3) that estimates the theoretical local density of the population probability distribution $f(x_i)$ directly based on the sample.

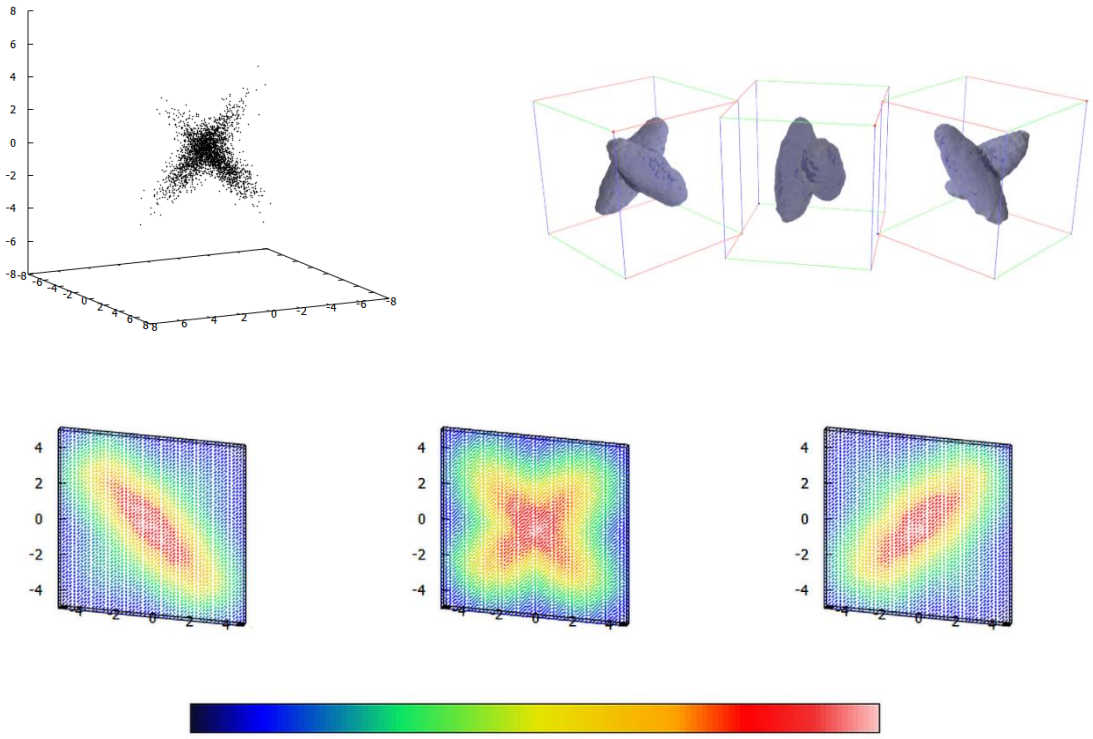


Figure 4: Raw data (a), views of PDF isosurface (b), and X-Z cross sections, colored by PDF (c). Description in Section 4.3. *Source: own work.*

$$f(x_i) = f_i \cong \frac{N_{const}}{N_0} \cdot \frac{1}{\Delta x_i} \tag{3}$$

The inverse histogram is an implementation of equation (3), via equation (4), taking into account the distance between ends of interval containing the subsample of constant size. The selected number N_{const} depends on the rank of the method 'k': $k \geq 1$, with $N_{const} := 2k$, if $k + 1 \leq i \leq N - k$. For the remaining points, the number of points per interval decreases with approaching the ends of the sample, so that the resulting set of values is equal in size to the input set (Eq. 4). The preservation of all measurement points is particularly important for small samples, which is typical in some experimental sciences, e.g. biology, econometrics, and management.

$$f(x) = \lambda \cdot \begin{cases} \frac{1}{y_2 - y_1} & , \ i = 1; \\ \frac{2(i-1)}{y_{2i-1} - y_1} & , \ i = 2, \dots, k; \\ \frac{2k}{y_{i+k} - y_{i-k}} & , \ i = k+1, \dots, N-k; \\ \frac{2(N-i)}{y_N - y_{2i-N}} & , \ i = N-k+1, \dots, N-1; \\ \frac{1}{y_N - y_{N-1}} & , \ i = N. \end{cases} \quad (4)$$

In the above formula $f(x_i)$ is an estimate of the local probability density in the sample at node 'i'. Both extreme points y_i are counted half way, so together they contribute to the numerator of the formula equal to '1' (Opiła (2018)), λ is normalization factor. Figure 1c shows a typical result obtained with this method.

The inverse histogram method has two significant weaknesses. First, it cannot be used for multidimensional data and second, it does not allow for calculating PDF estimates on an arbitrarily selected computational grid; as it is inherently pinned to measurement data. This limits the possibility of using it to estimate empirical probability distributions used in Monte Carlo simulations. For example, it cannot be used in the Characteristic Surface Method to estimate the empirical PDF required (Opiła (2020). The method presented below overcomes this problem.

The considerations presented hereby are illustrated in Fig.5. To simplify the discussion, the focus is on two-dimensional data, but the presented method was successfully applied to other dimensions (1, 3 and more), noting at the same time that the ability of visualizing the result for spaces more than three dimensions is limited.

By selecting a point on the plane (P or Q), one looks for the radius of the circle (*hypersphere*) that will accommodate selected number of measurement points – in Fig.5, these are seven points. The area of the circle, with radius r_P and r_Q, respectively, is the functional equivalent of the expression Δx_i for the inverse histogram (3). Finally, an estimate is obtained for the local probability density function, for example for point P: $f(P(x,y))$:

$$f\big(P(x,y)\big) = f_P \cong \lambda \frac{N_{const}}{\pi r_P^2} = \lambda \frac{N_{const}}{V_{hs}} \quad (5)$$

where V_{hs} denotes generalized volume of the hypersphere, eg. πr_P^2 for two dimensions. Furthermore, N_{const} is a fixed integer, selected as a parameter of the method, called *rank*, λ is a normalization factor calculated such that the total probability in the area equals one, and the point P is an arbitrarily chosen point (for example, node of the computation grid) in the area studied.

In areas where a larger than average number of measurements is concentrated (vicinity of point Q), which implicitly results from the higher value of the source PDF, the radius r_Q is smaller, resulting in a higher estimate of f_Q . In contrast, in areas where the PDF value is small (vicinity of point P), the value of r_P will be higher, producing a lower value of the estimate of f_P (see Fig.5). In practice, the entire algorithm can be presented in steps:

For each point $P_{i,j}$ of the computational grid,

- *find the radius $r_{i,j}$ of a circle such that it contains exactly R points,*
- *calculate $f_{i,j}$ from formula (5),*
- *perform normalization of the estimated PDF on the computation grid,*
- *perform visualization and any other required analyses.*

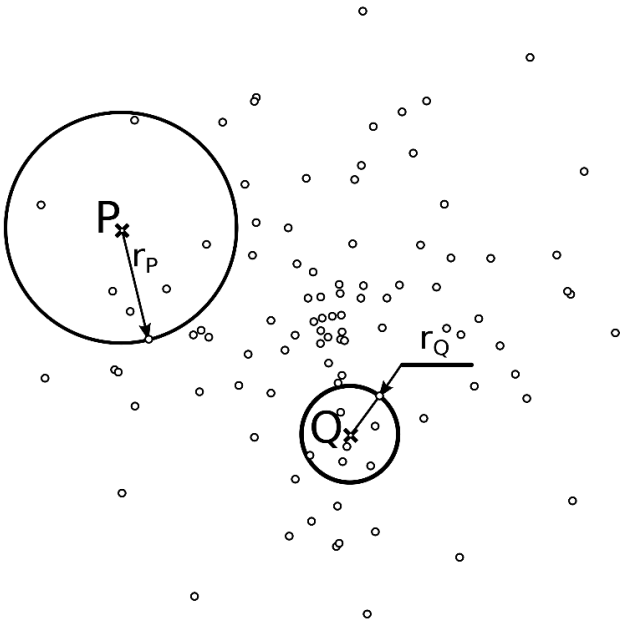


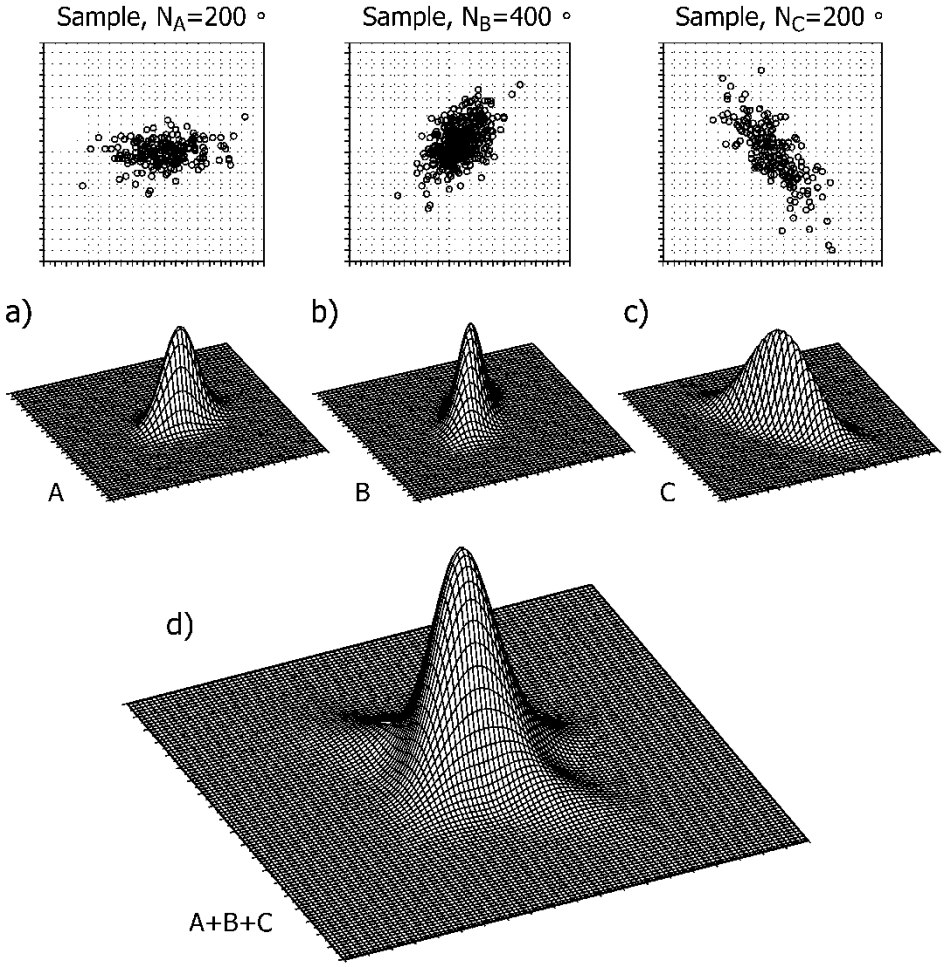
Figure 5: The aKDE method scheme. The distribution estimation is calculated at arbitrary selected points P and Q (marker '×'). Circular counting zones are shown for the rank R=7 (7 points). The location of the data points is marked with tiny circles. The radii of the zones are r_P and r_Q, respectively

Source: own study

The method requires an algorithmic approach; thus its working name is 'algorithmic KDE' (aKDE). For testing purposes, the aKDE algorithm was implemented using Python 3.13.1 and the numpy library. Although the aKDE algorithm may be implemented for any number of dimensions (1, …, N) without substantial changes, visualization of the obtained results is basically possible for dimensions from one to three. Thus, the paper focuses on the analysis of the two-dimensional case, ignoring the one-dimensional case as obvious, and the three-dimensional case as requiring a separate introduction (see Opiła and Pełech-Pilichowski (2018)). However, examples of 1D and 3D implementation will be presented. Accordingly, the next section discusses the test calculations using the aKDE method for two dimensions as a fully representative example.

Results

To ensure the absence of autocorrelation in pseudorandom number sequences and control covariance between data vectors, sampling was carried out using NumPy built-in method numpy.random.default_rng.normal(). The samples were obtained by random drawing from a mixed PDF of three independent components.



Two Dime

Figure 6: Mixed probability distribution used in the tests. Top row (a-c): drawn sample above, theoretical distribution – below. Bottom row: theoretical, mixed probability density function (d). Values of parameters of component distributions $((\sigma_X, \sigma_Y, \mu_X, \mu_Y)_{1|2|3})$ are included in the Table 1. *Source: own study*

Table 1: Parameters of the components of the test distribution in dimensionless units [u.d]. Parameters of the mixed PDF assumed in numerical experiments. The random sample contained a number of samples from each of the distributions in proportion depicted in column "Relative share".

LP.	σ_X [d.u.]	σ_Y [d.u.]	μ_X [d.u.]	μ_Y [d.u.]	ρ [d.u.]	Relative share [%]
1	1.66	1.06	6.0	5.0	0.27	25
2	1.41	1.74	5.0	6.0	0.58	50
3	1.91	2.52	5.0	5.0	-0.78	25

Source: own study

For a given sample, the result of the aKDE algorithm depends on four variables, listed in order of importance: the rank of the method R, the number of nodes in the computational grid G, the sample size N and the dimensions of the mapped area D. This set of parameters can be denoted using Pythonic *tuple* notation, for example S_i = (10,20,800,20) for $R = 10$, $G = 20$, $N = 800$, and $D = 20$. The complete test range should therefore contain different

tuples of these four quantities, which, in turn, generates $S_L = S_R \cdot S_G \cdot S_N \cdot S_D$ tuples (where '$S_{[R \mid G \mid N \mid d]}$' is the number of levels tested of the relevant variable). For example, for five levels, S_L = (5,5,5,5) the number is 625 cases. However, it is possible to limit the research space to one fixed-mapped area. Three of the most characteristic parameters tuples (in the author's opinion) were selected for presentation and comparison, the dimensions of the mapped area fixed to 20×20 units. In order to compare the algorithm discussed with KDE, three values of the smoothing constant were selected. Due to the geometric scalability of the issue, the labels of the axis markers were intentionally omitted from the graphs. Table 2 summarizes the variants tested.

Table 2: Parameters selected for numerical tests. Smoothing constant of the basic KDE algorithm is denoted as 'd'. The paper discusses selected combinations of elements, listed in column order (N,R,G,d). Explanation of symbols in the text.

LP	N	R	G	d
1	200	10	10	0.2
2	600	40	50	0.8
3	1200	70	100	5.0

Source: own work

The first to be investigated (see Fig.7) was the comparison of the theoretical distribution with the estimation by KDE with the Gaussian kernel for d = 1.0 and with the estimation by aKDE for R=40. The upper right corner shows sampled data points. The white circles represent the location of the $M_i = (\mu_X, \mu_Y)_i$ points for the components of the theoretical distribution (see Table 1). The arrows indicate the supposed relationship between the M_i points and the peaks in the graph made using the aKDE technique. Both methods fail in specific regions. KDE tends to show nonexistent peaks far from the maximum of the PDF, whilst aKDE shows distinct spikes in the central region of the maximal value of mixed PDF.

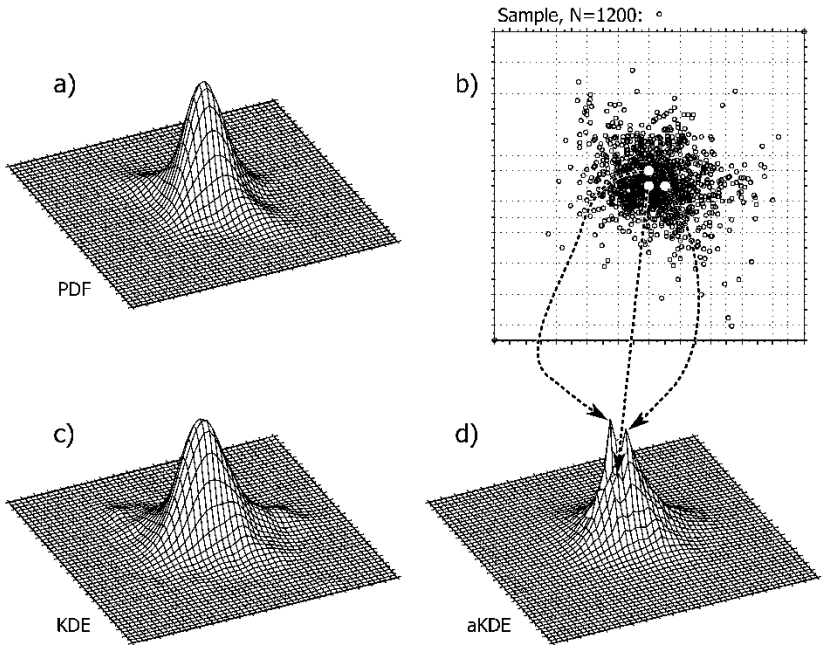


Figure 7: Comparison of the theoretical distribution (a), with the KDE estimation using the Gaussian kernel (b), for d=1.0 and with the estimation using aKDE for R=40, (d). The upper right corner shows the randomly selected data points (b). The white circles represent the location of the $M_i = (\mu_X, \mu_Y)_i$ points for the components of the theoretical distribution (see Table 1). The arrows indicate the supposed relationship between the M_i points and the peaks on the graph made using the aKDE technique.

Source: own study

Secondly, the dependence of the estimation result on the smoothing constant (bandwidth) 'd' (KDE) and the rank of the method 'R' (aKDE) with a constant 'N', N=1200 points and the size of the computation grid G (50×50 nodes) was investigated. The results obtained are presented in Fig.8. As can be seen, low values of the constants result in excessive fluctuations in the graph, with KDE generating a more 'noisy' graph (pair: Fig.8b vs. Fig.8e). At intermediate values, the KDE estimate approaches the theoretical distribution in the area of low PDF function values (in fact, strong 'noise' is still observed) – however, at the same time, it loses resolution near the maximum of the distribution (pair: Fig.8c vs. Fig.8f), smoothing it excessively. At the same time, aKDE maintains resolution throughout the range of PDF variability. A continued increase in both parameters leads to a complete blurring of the KDE estimate and only a slight reduction in resolution in the entire range of PDF variability in the aKDE method (pair: Fig.8d vs. Fig.8g). It can be stated that aKDE is more resistant to an incorrect choice of method rank than KDE. This is important because for both methods there is no objective criterion of the quality of the approximation. Despite that, the computed data may be sufficient for the least squares fitting.

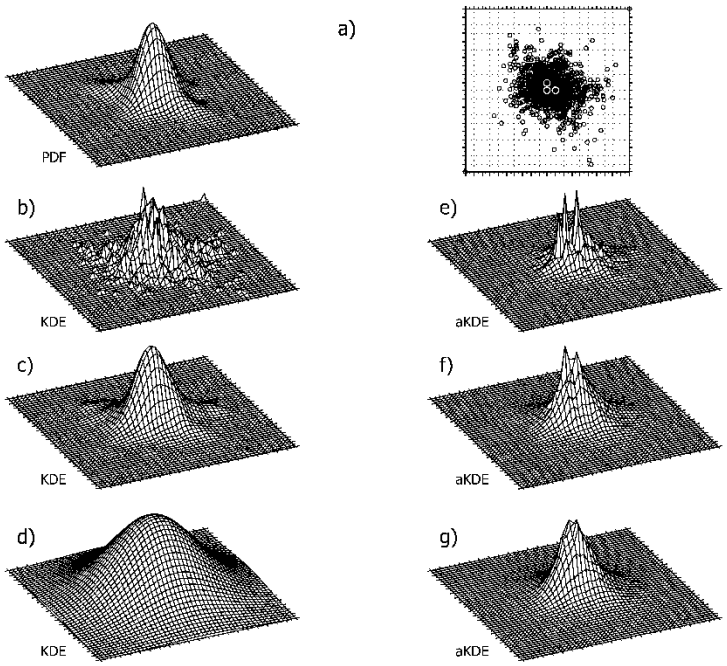


Figure 8: Comparison of the results of the KDE calculations for d=0.2; 0.8; 5.0 (b, c, d) respectively and aKDE, R=10,40,70, (e, f, g) – respectively. In the top row, for comparison, the theoretical distribution and the sample taken is shown (a). Assumed K=50, N=1200

Source: own study

For the third test, the parameters were set to N=1200 points, d=0.8 and R=40 while changing the density of the computational grid G. Three levels of the variable were chosen: 10, 50 and 100 nodes. As can be seen in Fig.9, both methods react similarly to the change in grid size, gradually increasing the resolution of the estimate. However, for aKDE, the shape of the estimated PDF remains spiky. Probably this results from an incorrect rank of the method R.

The last test was to examine the dependence of the estimate on the sample size while maintaining constant values of R=40, K=50, d=0.8. The result was analyzed for the values of N = 150, 600, and 1200 points (Fig.10). For the lowest sample size, in the case of KDE, artifacts are observed in the area with a low PDF value. Meanwhile, in the aKDE method, such artifacts do not occur. It should be recognized that the aKDE method can obtain a better approximation of the PDF value regardless of the sample size. This is a particularly important conclusion in the case of analyses conducted, often out of necessity, on small samples. To verify reliability of the computed PDF approximation, a direct comparison of the source PDF (surface) and the computed one (points) is shown in Fig.11.

Finally, a simple analysis of the computational complexity of the aKDE method was carried out. Due to the size of the computation grid, this complexity is $O(K \cdot w)$, where K is the number of grid nodes and w is the dimension of the space. However, the complexity due to the size of the test sample N is not so obvious. Theoretically, it should result directly from the computational complexity of the sorting algorithm – most often it will be a high performance Timsort(), with a computational complexity of $O(N \cdot log(N))$ in the worst case. However, based on the calculations performed for the wide range of sample sizes N, the complexity of the type $O(N)$ was observed (Table 3). This issue requires further in-depth studies.

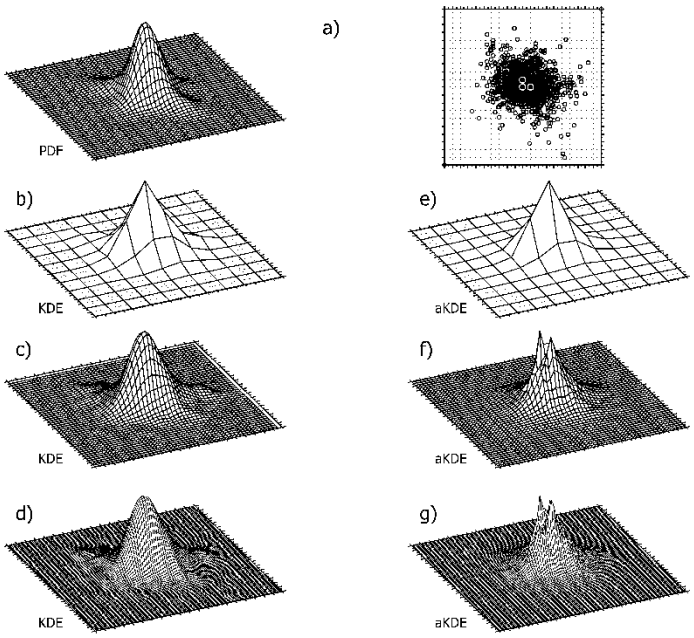


Figure 9: Comparison of the results of the KDE calculations for d=0.8 and the G=10; 50; 100 calculation grid (b, c, d – respectively) and aKDE, R=40 and the G=10; 50; 100 calculation grid (e, f, g – respectively). In the top row it is shown for comparison the theoretical distribution and the sample taken (a) Sample size N=1200.

Source: own study

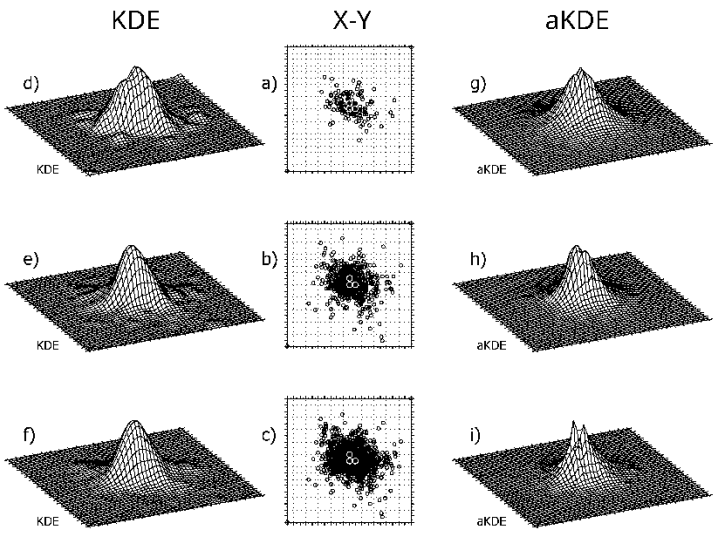


Figure 10: Comparison of the results of calculations using the KDE (*d*=0.8) and aKDE (*R*=40) methods for the sample size *N*=150, 600, 1200 (KDE: d, e, f; aKDE: g, h, i – respectively). The middle column (a, b, c) illustrates the sample taken (scatter or X-Y plot) for *N*=150; 600; 1200

Source: own study

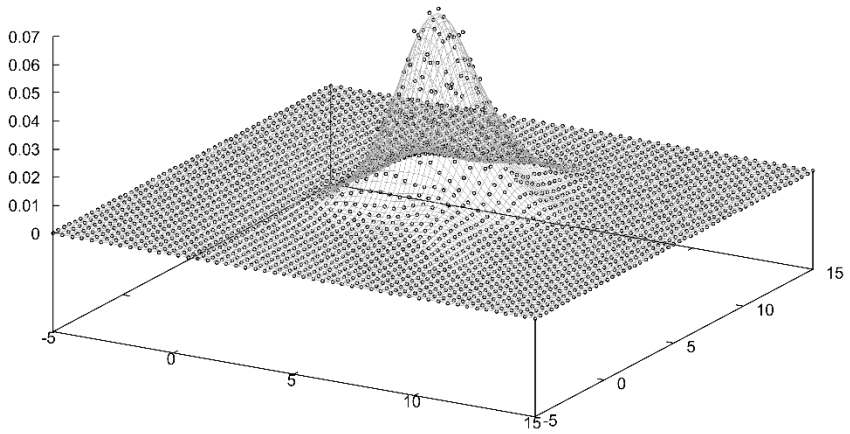


Figure 11: Comparison of original PDF (surface) with computed PDF (points). As can be seen shape of PDF is reproduced quite well.

Source: own study

Table 3: Approximate calculation time t[s], for R=20, K=50 as function of sample size N. Measurements done using time.perf_counter() Python function (quad-core Intel i5 processor, 2.4 GHz, Windows 10/64). Each presented time is average of computation times for five separate and independent samples, computed several times in order to eliminate random deviations.

LP	Sample size	t [s]	σ [s]
1	40	0.136	0.003
2	160	0.559	0.011
3	200	0.702	0.021
4	800	3.028	0.035
5	3200	11.575	0.054
6	12000	45.870	0.550
7	30000	114.430	1.097
8	50000	192.294	0.578

Source: own work

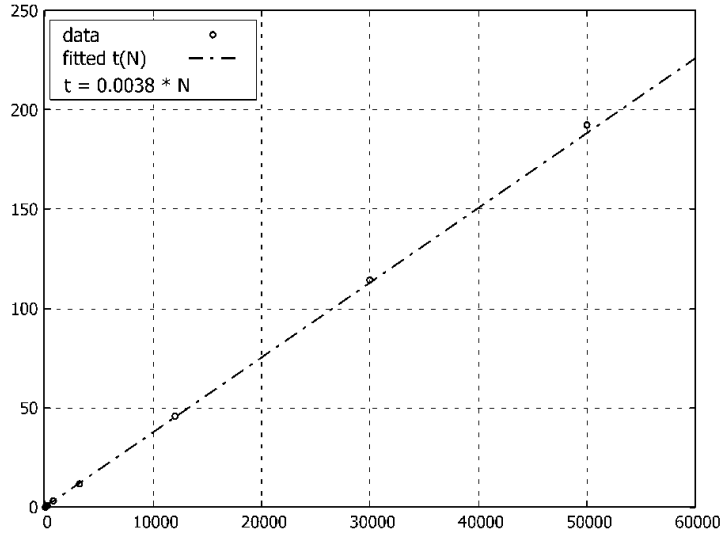


Figure 12: Computational complexity of the aKDE algorithm depending on the sample size N (see Table 3)

Source: own work

One Dimensional Case

The aKDE algorithm may be ported with only minor corrections to the one-dimensional case. In Fig. 13, the comparison of the theoretical complex PDF (dark gray line, light gray area) having three components (dotted black line) is shown with an aKDE estimate (black small circles) and the Marquardt-Levenberg Levenberg (1944), Marquardt (1963), fit to the data (solid red line). This result was obtained on a sample of 4000 data points using gnuplot (2024).

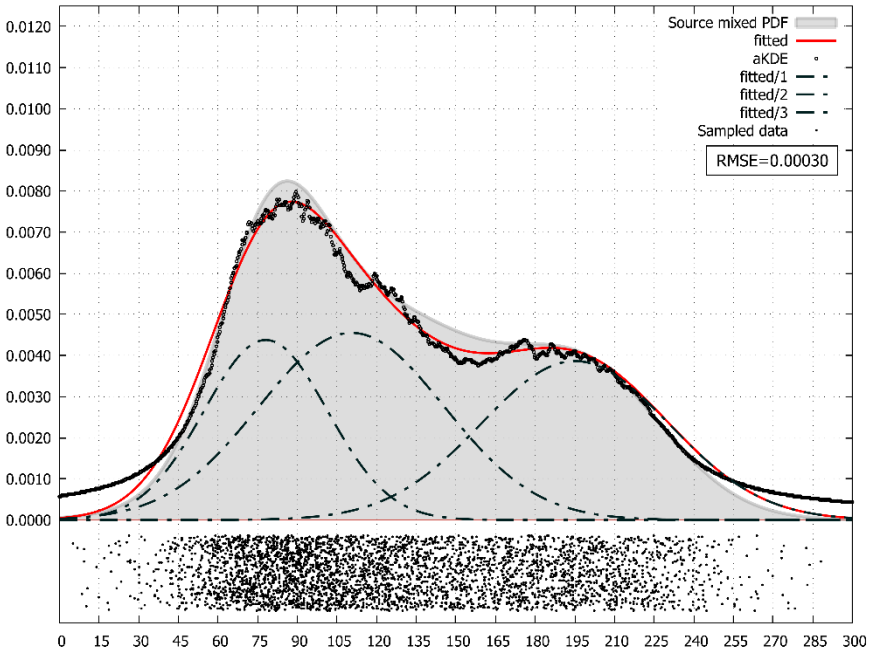


Figure 13: Example of PDF approximation in 1D case.

Source: own work

Three-dimensional case

Porting the aKDE to three dimensions requires only minor corrections to the computer code but important changes to the visualization schema. The example of results obtained in this work is presented in Fig.4. The Fig. 4a shows a cloud of sample points. The visualization of computed estimate of PDF is given in two ways. Figure 4b shows the isosurface for the Mahalanobis distance d_M = 3.0 (3σ) Mahalanobis (1936), shown in three views. This surface should cover 99.7% data points. There is a visible similarity between the shape of the cloud of raw points and the rendered isosurface. In this case POV-Ray raytracer (POVRay (2013)), enhanced with ScPovPlot3D (ScPovPlot3D (2024)) visualization package, was used. Figure 4c shows, colored by the value of local estimate of PDF, cross sections passing by centers of component distributions perpendicularly to the Y axis or parallel to X-Z plane (left and right) and the third one (central) through the origin of the coordinate system.

Conclusions

To sum up the analyzes performed, it is necessary to point out both positive and weak points of the method discussed above. In case of aKDE, among the strengths, it is necessary to mention the relatively simple mathematical formalism derived directly from the first principles, which cannot be said about techniques based on the KDE kernel smoothing function. Due to simplicity of the aKDE, the implementation of the method fits into a dozen or so lines of a Python script. In particular, it allows one to write plugins for programs using this algorithm. The analyses carried out indicate that aKDE is relatively resistant to incorrect parameter values, in other words, the degradation of fit quality is slow. At the same time, it should be noted that classic KDE techniques tend to either over-smooth or over-sharpen the tested sample. As shown, over sharpening and over smoothing can occur even within the same smoothing constant 'd', depending on the data region. The weaknesses of the aKDE method include the relatively high computational complexity resulting from the need to search for 'R' data points closest to the observation point. This leads to computational complexity dominated by the computational complexity of the applied sorting algorithm. In the case of using Timsort(), it is $O(N \cdot log(N))$ in the worst case. Luckily, aKDE, like the classic KDE, allows for massive parallelization, which in turn is a real advantage of both methods. In the tested cases, a linear dependence of the computational time on the sample size was observed up to 50000 data points, which may be due to the high degree of randomness of the drawn samples. It should be noted that aKDE, as a new method, has not been tested as thoroughly as competing methods, but it is difficult to treat this as its disadvantage. As shown in the work, imaging the dispersion of data using a multidimensional classic histogram can only play an auxiliary, preliminary role, preceding the actual analysis. This is due to the lowest computational

complexity of the methods tested here, which makes it actually fast. Despite that, the classic 2D histogram method is useless as an analytical tool.

The bottom line is that the aKDE method is a valuable alternative to classic KDE in data analysis, sometimes providing information that is not available using other multidimensional data visualization techniques.

Acknowledgment

The registration fee was funded under subvention funds for the AGH University of Krakow, Poland

Endnotes

[1] $K_F\ (x) = \left(1 + \ exp\left(\frac{\frac{x}{d} - \mu}{T}\right)\right)^{-1}$, where '$d$' is a bandwith, Opiła (2018).

References

- Doane, D. (1976). Aesthetic frequency classification. American Statistician 30, 181–183,
- Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. Quarterly of Applied Mathematics 2 (2), 164–168,
- Mahalanobis, P. C. (1936). On the generalized distance in statistics. Proceedings of 345 National Institute of Sciences (India) 2 (1), 49–55,
- Marquardt, D. (1963). An algorithm for least-squares estimation of nonlinear parameters, SIAM Journal on Applied Mathematics 2 (11), 431–441, doi:10.1137/0111030,
- Opiła, J. (2018). Wizualna analiza dyspersji danych, Wspomaganie zarządzania z wykorzystaniem technologii IT, Kiełtyka L. (ed), Wydawnictwo Politechniki Częstochowskiej, Częstochowa,
- Opiła, J. (2019). Role of visualization in a knowledge transfer process, Business Systems Research 10 (1), 164–179,
- Opiła, J. (2020, Sep.), Employing of extended characteristic surface model for forecasting of demand in tourism. ENTRENOVA - ENTerprise REsearch InNOVAtion 6 (1), 60–73,
- Opiła, J. and T. Pełech-Pilichowski (2018), Visualization of irregular datasets using kernel density estimation function, (ed) K. Skala, MIPRO 2018 proceedings, 41st International Convention, pp. 217–222. ISSN 1847-3946), e-ISBN: 978-953-233-096-0, DOI: 10.23919/MIPRO.2018.8400036.
- POV-Ray (2013, November), Persistence of vision (tm) raytracer, [Online] URL; https://www.povray, org/. ver.3.7, [Retrieved: 2025-05-29],
- Scott, D. (1979). On optimal and data-based histograms. Biometrika 66, 605–610,
- ScPovPlot3D (2024, June), [Online] URL https://github.com/JustJanush/Plot3Dv4. GPL, v.4.0, [Retrieved: 2025-05-29],
- Statistica (2016). [Online] URL: https://www.statsoft.pl/, [Retrieved: 2025-05-29],
- Sturges, H. (1926, Mar.), The choice of a class interval, J. American Statistical Association 21 (153), 65–66,
- Várkoly, L. (2017). Power of visualization im modern teaching/training process, Wybrane zagadnienia zarządzania współczesnymi przedsiębiorstwami, Vol. 1, pp. 80–90. wyd. P.Cz., Częstochowa,
- Wiliams T., Kelley C., et al. gnuplot (2024, May). Gnuplot 6.0.1: an interactive plotting program, [Online] URL: https://gnuplot.sourceforge.net/ by Thomas Williams and Colin Kelley et al. [Retrieved: 2025-05-29],