Multi-Resolution Data Processing: It is Necessary, It is Possible, It is Fundamental

S. A. Starks[1], V. Kreinovich[1], and A. Meystel[2]
[1]NASA Pan-American Center for
Environmental and Earth Studies (PACES)
University of Texas at El Paso, El Paso, TX 79968
emails {sstarks,vladik}@utep.edu
[2]National Institute of Standards and Technology
Gaithersburg, MD 20899

ABSTRACT

Experience shows that many data processing problems are difficult to solve, and some of these problems have even been *proven* to be computationally intractable. Human experts successfully solve many such problems by using a hierarchical, *multi-resolution* approach. These multi-resolution methods are, in several cases, provably optimal. However, due to the computational intractability of the problem itself, the multi-resolution approach can only work if the systems that we are analyzing are themselves hierarchical. We show that, first, due to (inevitable) measurement inaccuracies, an arbitrary input data is *consistent* with the hierarchical model, and second, that in many cases, the actual physical world is indeed fundamentally hierarchical.

Since traditional statistical methods have been designed primarily for non-hierarchical models, their direct application to multi-resolution data processing can lead to biased estimates. On a simple example, we show how these methods can be corrected to avoid this bias. Surprisingly, the analysis of this problem leads to new unexpected *symmetries*.

KEYWORDS: *multi-resolution data processing, granularity, computational intractability, wavelets, fractals, satellite image, semiotics, environmental studies, Schroedinger's paradox*

1. DATA PROCESSING IS DIFFICULT

Data processing is difficult: an empirical fact. From the engineering viewpoint, the main problem with data acquisition and processing is to manage to *acquire* the data. To launch a successful space mission to other planets is indeed an extraordinary engineering achievement. However, even for such missions, the pure volume of acquired data is so huge that processing all this data becomes a very difficult task.

Computers become faster and faster, new algorithms are designed, and therefore, data processing becomes faster and faster; however, at the same time, this same progress leads to better data acquisition devices that drastically increase the amount of raw data, and to new ideas of what additional information we can extract from the old data. A typical NASA-related example: the University of Texas at El Paso, together with Jet Propulsion Lab, is currently analyzing the data from Mariner missions to find relativistic effects on the spaceships' trajectories.

The complexity of data processing is a difficult problem for many different application areas, but is especially difficult for areas in which it is relatively easy to get new data. *Environmental* and earth studies are one of such areas: lots of relatively *easily accessible data* come from satellites, and processing this data becomes more and more difficult.

This problem is going to become even more acute in the nearest future: Indeed, currently, environmental-related satellites use a few frequencies (no more than 10). From the resulting measurement results, we only get a small sample of the spectrum, and from this small portion, it is often difficult to tell one type of terrain from another (or even from a cloud formation). To get a better understanding of Earth features, NASA is currently planning to launch a series of new *multi-spectral* satellites that, for each point, will measure up to several hundred intensities instead of the usual few. This increase in data flow is definitely advantageous, but it makes data processing even more more complicated.

Data processing is difficult: theoretical results. Every once in a while, new algorithms appear that drastically decrease the computation time of different data processing problems. This continuous progress in algorithm design may lead to an (over)optimistic viewpoint that sooner or

later, *ideal* algorithms will emerge that will perform all the data processing tasks in *real time* (i.e., the current data will be processed by the time when new data will arrive).
Alas, this optimism is unfounded: numerous theoretical results show that in the general case, data processing problems are *computationally intractable* (or, to use the precise term from the theory of computing, *NP-hard*). This is true for general data processing (see [8] and references therein), for the problems of reconstructing the past [1,2], for problems of quantum mechanics [10] and space-time geometry [7], etc.
Crudely speaking, these results mean that for any algorithm that solves a data processing problem, there exist possible data on which this algorithm takes *exponentially long* time, i.e., time that grows as 2^n, where n is the length of the input (measured, e.g., in bits). Already for reasonably small n (e.g, for $n \approx 300$), the required time exceeds the lifetime of the Universe. In short, such problems are indeed intractable.

The problem. The problem is: *how to process real-life data?*

2. MULTI-RESOLUTION DATA PROCESSING IS NECESSARY

How do we humans solve this problem? Enter the idea of multi-resolution data processing. There are many data processing problems in which human experts are much better than computers, image processing one of them. To be more precise:

- If we have a *small* amount of data, e.g., a blurry noisy image, a human eye will only see the noise, while a computer can perfectly well filter this noise out and get, e.g., all these nice pictures that we see coming from the Hubble telescope.
- On the other hand, if we have a *huge* amount of data, e.g., if we need to identify a face on a photo or a geological pattern on satellite image, then a trained human eye does it in no time, while supercomputers often take forever or even sometimes fail.

How do we humans do it? Definitely not because our brain is faster than a computer: its main processing elements (neurons) have a processing time of 10–100 milliseconds. The main reason why we can do this job is because we do not store and we do not process the pixel-by-pixel image as a computer would normally do. Instead, we store and remember image in a very compressed form, usually, as a small collection of standard images described, usually, in *semiotic* form, i.e., by *words* and *symbols* (mental or real). Moreover, this description is usually *hierarchical*, *multi-scale*: first, we remember and describe the "big picture" (main features), then we go into more details (i.e., into a somewhat smaller scale), etc.
Since we humans use this idea, and use it successfully, it is desirable to make computer programs use this idea as well.

Wavelets: a mathematical representation of the idea of multi-resolution data processing. One of the most practically successful formalizations of the idea of multi-resolution data processing is the *wavelet* technique (see, e.g., [13]). Crudely speaking, a wavelet transform decomposes the original image into "sub-images" that correspond to the large-scale details, medium-scale details, etc. So, we can store and process these sub-images instead of the entire original image.

Multi-resolution methods are indeed optimal. Many people have heard about wavelets, and it is reasonably well known that in many application areas (e.g., in data compression) wavelet-based methods are often indeed better than more traditional techniques. What is less known is that wavelet-based methods are not simply biologically motivated and *empirically* good: there are mathematical results that show that in many problems, wavelet-based methods are indeed *optimal*. We will mention two such results:

- The first result is closer related to human way of data processing: it tells that among all possible neural networks, neural networks that use *wavelet-type* activation functions have (asymptotically) the best approximation property [9].
- The second result is directly related to data processing, namely, to image processing: it shows that in a certain class of problems (like the problem of automatically detecting whether a surface mounted device is correctly mounted on a chip) wavelets are indeed the best data compression method [6]. This second result is not just an *asymptotic* optimality result: it has actually led to successfully wavelet-based image processing results [3,5].

Multi-resolution methods may be the best of the possible ones, but how come they are good? The very fact that multi-resolution methods are the *best* (i.e., the fastest) of all possible methods of solving data processing problems does not invalidate the above-cited pessimistic result that this problem is computationally intractable. In other words, *theoretically*, one can describe possible combinations of data for which these methods will not work.
However, both our own experience (as human experts we use multi-resolution methods) and the experience of data

processing algorithms that use multi-resolution techniques show that these methods are practically very feasible, in other words, that these horror worst-cases practically do not happen in real life. The question is: why? Is nature *designed* in such a way that these methods always work, or simply we were lucky so far and bad cases will still appear in the future?
Our answer is optimistic: yes, nature is designed in this manner.

3. MULTI-RESOLUTION DATA PROCESSING IS POSSIBLE AND IS FUNDAMENTAL

Two explanations. We will give two explanations why nature is designed in this way:

- Our first explanation will say, crudely speaking, that even when nature can form arbitrarily complicated images and data strings, the inevitable presence of *noise* and *measurement errors* makes every observation compatible with a hierarchical model (i.e., with a model that can be handled by multi-resolution techniques).
- Our second explanation is that not only the *approximate* image of nature is hierarchical, but the nature itself is granular and hierarchical.

First explanation: Tsirelson's theorem. [15] Tsirelson noticed that in many cases, when we reconstruct the signal from the noisy data, and we assume that the resulting signal belongs to a certain class, the reconstructed signal is often an *extreme* point from this class. For example, when we assume that the reconstructed signal is monotonic, the reconstructed function is often (piece-wise) constant; if we additional assume that the signal is smooth (one time differentiable, from the class C^1), the result is usually one time differentiable but rarely twice differentiable, etc.
Tsirelson provides an elegant *geometric* explanation to this fact: namely, when we reconstruct a signal from a mixture of a signal and a Gaussian noise, then the *maximum likelihood*estimation (a traditional statistical techniques) means that we look for a signal that belongs to the priori class, and that is the closest (in the L^2-metric) to the observed "signal+noise". In particular, if the signal is determined by finitely many (say, d) parameters, we must look for a signal $\vec{s}=(s_1,\ldots,s_d)$ from the a priori set $A\subseteq R^d$ that is the closest (in the usual Euclidean sense) to the observed values $\vec{o}=(o_1,\ldots,o_d)=(s_1+n_1,\ldots,s_d+n_d)$, where n_i denotes the (unknown) values of the noise.
Since the noise is Gaussian, we can usually apply the *central limit theorem* and conclude that the average value of $(n_i)^2$ is close to σ^2, where σ is the standard deviation of the noise. In other words, we can conclude that $(n_1)^2+\ldots+(n_d)^2\approx d\sigma^2$. In geometric terms, this means that the distance $\sqrt{\sum(o_i-s_i)^2}=\sqrt{\sum n_i^2}$ between $\vec{s}$ and $\vec{o}$ is $\approx \sigma\sqrt{d}$. Let us denote this distance $\sigma\sqrt{d}$ by ε.
Let us (for simplicity) consider the case when $d=2$, and when A is a convex polygon. Then, we can divide all points p from the exterior of A that are $\varepsilon-$close to A into several zones depending on what part of A is the closest to p: one of the *sides*, or one of the *edges*. Geometrically, the set of all points for which the closest point $a\in A$ belongs to the *side* e is bounded by the straight lines orthogonal (perpendicular) to e. The total length of this set is is therefore equal to the length of this particular side; hence, the total length of all the points that are the closest to all the sides is equal to the *perimeter* of the polygon. This total length thus does not depend on ε at all. However, the set of all the points at the distance ε from A grows with the increase in ε; its length grows approximately as the growth of a circle, i.e., as const$\cdot\varepsilon$. When ε increases, the (constant) perimeter is a vanishing part of the total length. Hence, for large ε, the fraction of the points that are the closest to one of the sides tends to 0, while the fraction of the points p for which the *closest* is one of the *edges* goes to 1.
Similar arguments can be repeated for any dimension. For the same noise level σ, when d increases, the distance $\varepsilon=\sigma\sqrt{d}$ also increases, and therefore, for large d, for "almost all" observed points $\vec{o}$, the reconstructed signal is one of the extreme points of the a priori set A.
Much less probable is that the reconstructed signal belongs to the 1-dimensional face of the set A, even much less probable that s belongs to a 2-D face, etc.
The main *methodological consequence* of this result is that even when the actual state space is *continuous*, when we determine the state from measurements result, we inevitably obtain (most often) one of the *discretely many* states. On the large-scale level, we get one of the few *clusters*. When we add new measurements and thus, get to the next level, each original cluster sub-divides into new clusters, etc., so that we get a *hierarchical* structure.

Comment: Schroedinger's paradox and other methodological applications of Tsirelson's result. In *quantum mechanics*, this result explains why pure states (extremal points) are much more frequent that mixed ones; in *history*, it explains why there are finitely many types of social organization; in *logic*, it explains why in spite of the clearly fuzzy character of most human reasoning, binary logic describes most of this reasoning pretty well, etc. In particular, it explains the famous "cat" paradox proposed by E. Schroedinger, one of the founding fathers of quantum mechanics,
In *classical physics*, it is assumed that for each state of a physical system, every property is either true or false. For example, a particle is either located in a certain interval

of space coordinates $[x - \Delta, x + \Delta]$, or it is not located inside this interval. In *quantum mechanics*, in addition to the states in which a particle is located within this interval, and to the states in which the particle is definitely outside it, there are states in which *some* measurements of the coordinate will lead to results *within* the interval, and *some* to the results *outside* this interval. In such states, we cannot say that a statement "the particle is located in the given interval" is true or that this statement is false; at best, we can determine the *probability* of the "yes" answer. (To describe such unusual "truth value", quantum logic has been introduced.)

States with unusual "truth values" are not an exception, but rather a *general* rule in quantum mechanics: e.g., for every two states ψ and ψ' with certain values $\lambda \neq \lambda'$ of a measured quantity, there exists a state called their *superposition* in which the value of this quantity is no longer certain. (In the standard formalism of quantum mechanics, where states are described by vectors in a Hilbert space, superposition is simply linear combination.)

Such superposition state is easy to generate. Schroedinger has shown that this *superposition principle* seemingly contradicts our intuition: indeed, suppose that we have a cat in a box, and a light-controlled rifle is aimed at the cat in such a way that a left-polarized photon would trigger the rifle and kill the cat, while the right-polarized photon would keep the cat alive. If we send a photon with a circular polarization (that is, according to quantum mechanics, a superposition of left- and right-polarized states), we would get (due to the *linear* character of the equations of quantum mechanics), the *superposition* of the states resulting from using left- and right-polarized photons. In other words, we will get a *superposition of* a *dead and alive* cat states. This is, however, something that no one has ever observed: for macroscopic objects (cats included), an object is either dead or alive. Tsirelson's result explains why such non-extremal states are indeed difficult to observe.

Second explanation: Fractal (hierarchical) structure of the Universe [11]. At first glance, the Universe as a whole seems to be uniform: in whatever direction we look, there are, on a large scale, approximately the same amount of galaxies. However, as early as the 19th century, Olbers showed (in his famous paradox) that this impression is false: If indeed the matter was homogeneously distributed, then the total brightness of all the stars located at distances between R and $R+\Delta R$ would be proportional to the volume $R^2 \cdot \Delta R$ of the corresponding spherical segment. Since the brightness dims with distance as R^{-2}, the resulting Earth-observed brightness would be the same irrespective of R, and the total brightness caused by all the stars would be *infinite* or at least very large. As a result, argued Olbers, it would be as bright at night as it is at daytime. The only way to avoid this paradox and to retain the observable homogeneity with the observable night darkness is to take into consideration that the Universe is *hierarchical*: stars form galaxies, galaxies form galaxy clusters, etc. The larger scale we go it, the less space is taken by matter, and the more by vacuum. The resulting *fractal* description of matter distribution is indeed consistent.

Olbers paradox was the first but not the only occurrence of meaningless infinity in seemingly meaningful physical equations. Such infinities consistently emerge in field theory, both classical and quantum. An interesting mathematical fact that is that if we consider field theories in space-time of arbitrary dimension d, then infinities only occur for (small) integer d, in particular, for the physically meaningful $d = 4$, but they do not occur for $d = 4 - \varepsilon$ for a small $\varepsilon > 0$. Currently, this idea is used as a formal trick, to *compute* the physical quantities by using fractal dimensions, but it is reasonable to conclude, from this result, that the *actual* dimension of space-time is fractal, i.e., that space-time indeed has a fractal structure [4]. Since space-time is also homogeneous, this conclusion means, crudely speaking, that not all points from the 4-D continuum describe events from the actual space-time, but that these events actually form a hierarchical structure.

4. MULTI-RESOLUTION DATA PROCESSING: PROBLEMS AND CHALLENGES

Traditional statistical methods are based on non-hierarchical data. Traditional statistical methods treat the entire data processing as a *single* process, going from input (initial data) to the output (classification or values of different quantities). To be more precise, there exist *multi-step* methods, but these are methods that simplify the computations at the expense of the artificially added hierarchical structure, and *not* by using the actual hierarchical structure.

Traditional statistical methods and multi-resolution data processing: a problem. Since traditional statistical methods are oriented towards one-step data processing, when we have a multi-resolution, multi-stage processing, we apply the traditional statistical methods to each stage separately, as if at each stage, we start with the raw data, and return the final results of data processing.

In reality, after, e.g., the first step of data processing, we do not have raw data any more, we have *pre-processed* data; due to this pre-processing, the error probability distribution for pre-processed data is different from a typical error probabilities for raw data, and therefore, strictly speaking, traditional methods are no longer applicable.

This problem is very urgent for processing environmental data, especially for processing earth-based environmental data (that usually supplements the data coming from satellite imaging). This data, usually, does not come directly from measurements: the raw measurement results are processed and generalized; then measurements corresponding to a certain small area are processed together, etc.; quite a few levels of data processing pass before we even get the data.

2-D example. How does this multi-stage processing affect the results? Let us give a simple illustrative example. We have already mentioned (when describing Tsirelson's result) that, from the geometric viewpoint, standard data processing techniques correspond to finding the point $\vec{s}$ from the a priori set A that is the closest to the observation point $\vec{o}$ (the closest in the sense of either the standard Euclidean distance or of its multi-dimensional analogue).

Let us start, for simplicity, with a 2-D case ($d = 2$), and let us consider an a priori ellipse

$$A = \{(x_1, x_2) \mid \frac{x_1^2}{A^2} + x_2^2 = R^2\}.$$

To get a point $\vec{s} \in A$ is the ultimate goal of data processing. At each intermediate step, we achieve this goal only *partly*, i.e., we get a intermediate point from some *larger* set; at each stage, this set gets smaller and smaller until finally, we get a point from the desired set A. It is natural to assume that these intermediate sets are also *ellipses* that are similar to the desired set A, with the only difference that they correspond to *larger* values R; from one step of data processing to the next, the value R gets smaller and smaller until we reach the desired value.

In other words, we start with a point $\vec{o}$ that comes from observations. On the first step of data processing, we find a point $\vec{s}^{\,(1)}$ from the ellipse $A^{(1)}$ (corresponding to the value $R^{(1)}$) that is the closest to $\vec{o}$. One the second step, we find the point $\vec{s}^{\,(2)}$ from the ellipse $A^{(2)}$ (corresponding to the value $R^{(2)} < R^{(1)}$) that is the closest to $\vec{s}^{\,(1)}$, etc. After N processing steps, we get a point $\vec{s}^{\,(N)}$ from the desired ellipse A. This point is the closest to the previous point $\vec{s}^{\,(N-1)}$ but, as one can easily see geometrically, this point is *not* necessarily *the closest* to the original observation $\vec{o}$ from all the points from A. In other words, the result of multi-stage processing algorithm is *different* from the desired point $\vec{s}$. How different is it? And how can we compensate for this difference?

To simulate the effect of a large number N of stages on the result of data processing, let us consider the limit of *infinitely many* stages. In this limit, instead of *finitely many* different ellipses $A^{(1)}, \ldots, A^{(N)}$ that correspond to decreasing values $R^{(1)} > \ldots > R^{(N)} = R$, we get a *continuous family* of ellipses that correspond to decreasing value of the parameter R. Similarly, instead of *finitely many* intermediate results $\vec{s}^{(1)}, \ldots, \vec{s}^{\,(N)}$ of data processing, we get a *continuous* family of points. Geometrically, this continuous family of points forms a *curve* $x_2 = f(x_1)$.

To describe this continuous process, let us describe how the "next" point on the curve (that describes different stages of data processing) is related to the "previous" one, i.e., to be more precise, let us describe a *differential equation* for this curve. Each point (x_1, x_2) on this curve belongs to a certain ellipse (with the value $R = \sqrt{x_1^2/A^2 + x_2^2}$). To get the "next" point on this curve, we consider a slightly smaller ellipse, with the parameter $R - \Delta R$, and take the point on that smaller ellipse that is the closest to the given one. The straight line that connects the original point with the "next" one is, in geometric terms, a *tangent* to the curve. It is well known from geometry that the straight line segment from any point to its closest point on any surface or curve (in particularly, on an ellipse) is *orthogonal* to this surface or curve (i.e., to the *tangent* to this surface or curve). Thus, at any point, the tangent to the desired curve is orthogonal to the tangent to the ellipse that this curve currently passes through.

The tangent to the ellipse $x_1^2/A^2 + x_2^2 = \text{const}$ can be obtained if we differentiate both sides of the ellipse's equation: $(2x_1/A^2)\, dx_1 + 2x_2\, dx_2 = 0$. Dividing both sides by 2, we get $(x_1/A^2)\, dx_1 + x_2 \cdot dx_2 = 0$. The orthogonal line to this tangent is, therefore, described by the equation $dx_1/(x_1/A^2) = A^2\, dx_1/x_1 = dx_2/x_2$. This differential equation can be easily integrated, leading to $c + a \cdot \ln(x_1) = \ln(x_2)$, where $a = A^2$ and c is an arbitrary constant, i.e., to $x_2 = C \cdot x_1^a$ (where we denoted $C = \exp(c)$).

We can use this equation to *correct* the effects of multi-stage data processing: namely, if we know the values (x_1, x_2) that correspond to several different stages of data processing, then we can:

- use the *least squares* method to find the parameters a and c from the equation $\ln(x_2) = c + a \cdot \ln(x_1)$;
- use the resulting formula $x_2 = \exp(c) \cdot x_1^a$ to reconstruct the original values x_1 and x_2; and then,
- use least square method again to find the point $\vec{s}$ on the ellipse A that is the closest to the original observations (x_1, x_2).

Multi-dimensional case. In a 2-D case, we get reasonably simple formulas. It turns out that in a more realistic multi-D case, the resulting formulas are only slightly more complicated. Indeed, if instead of *ellipses*, we consider *ellipsoids*

$$A = \{(x_1, \ldots, x_n) \mid \frac{x_1^2}{A_1^2} + \ldots + \frac{x_n^2}{A_n^2} = R^2\},$$

then a similar orthogonality condition means that a tangent to the curve (that represents the consequent intermediate results of multi-stage data processing) is orthogonal to the surface of the ellipsoid. This condition leads to a system of equations

$$\frac{dx_1}{x_1/A_1^2} = \frac{dx_2}{x_2/A_2^2} = \ldots = \frac{dx_n}{x_n/A_n^2},$$

from which we conclude that $\ln(x_i) = a_i \cdot \ln(x_1) + c_i$, i.e., that $x_i = C_i \cdot x_1^{a_i}$.

Surprising emergence of symmetries. An interesting side effect of our analysis is that the resulting curve has an unexpected *symmetry*: namely, if we change a unit in which we measure x_1 to a unit that is $\lambda > 0$ times smaller (i.e., if we replace x_1 by $\tilde{x}_i = \lambda \cdot x_i$), then we get exactly the same formulas for the relationship between x_i if we appropriately change the units for all other variables x_i. Moreover, the relationship $x_i = C_i \cdot x_1^{a_i}$ is the only possible relationship with this property.
This particular symmetry is very important (for numerous examples of using this and more complicated symmetries in computer science and data processing, see, e.g., [14]). The very fact that this important symmetry comes as a consequence of the hierarchical structure of data processing makes us believe that, maybe, *symmetry* in general, with all its important applications in physics and in other areas, *can be explained based on the granular* hierarchical *structure* of the Universe.

Acknowledgments. This work was supported in part by NSF under grant No. EEC-9322370 and by NASA under cooperative agreement NCCW-0089.

5. REFERENCES

[1] Alefeld, G., Koshelev, M., and Mayer, G. "Fixed Future and Uncertain Past: Theorems Explain Why It Is Often More Difficult To Reconstruct the Past Than to Predict the Future", in Proceedings of the NASA University Research Centers Conference, Albuquerque, New Mexico, February 16–19, 1997, pp. 23–27.

[2] Alefeld, G., Koshelev, M., and Mayer, G. "Why is it computationally harder to reconstruct the past than to predict the future?", *International Journal of Theoretical Physics*, Vol. 36, pp. 1709–1715, 1997.

[3] Carrillo, G., Cabrera, S.D., and Portillo, A.A. "Inspection of Surface-Mount-Device images using wavelet processing", in Proceedings of SPIE Applied Imagery Pattern Recognition, Washington D.C., October 1996.

[4] Finkelstein, A.M., and Kreinovich, V. "The singularities in quantum cosmology," *Astrophysics and Space Science*, pp. 73–76, 1987.

[5] Gallegos, J.M., Villalobos, J.R., Carrillo, G., and Cabrera, S.D., "Reduced-dimension and wavelet processing of SMD images for real-time inspection", In Proceedings of the IEEE Southwest Symposium on Images Analysis and Interpretation, San Antonio, April 1996.

[6] Kosheleva, O., and Brito, A.E. "Interval + Image = Wavelet", submitted to *Reliable Computing.*

[7] Kreinovich, V. "Spacetime isomorphism problem is intractable (NP-hard)," *International Journal of Theoretical Physics*, Vol. 30, pp. 1249–1257, 1991.

[8] Kreinovich, V., Lakeyev, A., Rohn, J., and Kahl, P. *Computational complexity and feasibility of data processing and interval computations*, Boston: Kluwer, 1997 (to appear).

[9] Kreinovich, V., Sirisaengtaksin, O., Cabrera, S.D. "Wavelet neural networks are optimal approximators for functions of one variable," in Proceedings of the IEEE International Conference on Neural Networks, Orlando, FL, July 1994, Vol. 1, pp. 299-303.

[10] Kreinovich, V., Vazquez, A., and Kosheleva, O. "Prediction problem in quantum mechanics is intractable (NP-hard)," *International Journal of Theoretical Physics*, 1991, Vol. 30, pp. 113–122, 1991.

[11] Mandelbrot, B.B. *The fractal geometry of Nature*, San Francisco: Freeman, 1982.

[12] Meystel, A. "Evolution of knowledge in large complex systems", in *Control Mechanisms for Complex Systems: Issues of Measurement and Semiotic Analysis*, Proceedings of the Workshop, Las Cruces, NM, 1996, pp. 231–254.

[13] Meyer, Y. *Wavelets algorithms and applications*, Philadelphia: SIAM, 1993.

[14] Nguyen, H.T., and Kreinovich, V. *Applications of continuous mathematics to computer science*, Dordrecht: Kluwer, 1997 (to appear).

[15] Tsirel'son, B.S. "A geometrical approach to maximum likelihood estimation for infinite-dimensional Gaussian location. I", *Theory Probab. Appl.*, Vol. 27, pp. 411–418, 1982.