Scientific Visualization, 2024, volume 16, number 5, pages 120 - 150, DOI: 10.26583/sv.16.5.09

Stochastic Semantics of Big Data (Parallel Computing and Visualization)

D. V. Manakov[1,A], P. A. Vasev[2,A]

Krasovsky Institute of Mathematics and Mechanics, Ural Branch of the Russian Academy of Sciences

[1] ORCID: 0000-0001-6852-8096, manakov@imm.uran.ru
[2] ORCID: 0000-0003-3854-0670, vasev@imm.uran.ru

Abstract

First of all the paper considers the problem of verification or formalization of the online visualization and parallel computing system from the point of view of dynamic systems as a development of the theory of computational complexity for random processes. Considering problems involving truly big data inevitably leads to the use of a block approach which is also used in both information theory and stochastic differential equations. As a natural metaphor the graph signals were chosen. This is a graph in nodes, of which a spectral function is defined in the examples considered this is a function of color (RGB), height or amount of data. In parallel computing, a block can be associated with a computing unit (processor) and consider the problem of entropy (performance) maximization. In the developed on-line visualization and concurrent computing system for geometric parallelization, it is possible to implement and compare a stationary random process (equiprobable messages implemented using broadcasting and mixins) and a steady-state random process (point-to-point messages), which have different analytical solutions. Together, this allows concluding that the proposed implementation of a stationary process has a certain novelty; in addition, it was intended to be more convenient for automated parallelization. The problems of automatic load balancing (interpolation problem) and optimal scalability of parallel computing (extrapolation problem) are also considered. Not much has been done in the field of visualization verification for example a mesh visualization has been proposed to be considered as a parameterized model of a white-noise random process. Of course, this work cannot be considered complete, but the direction that the authors called stochastic semantics is obviously promising.

The authors intend to take a close look at the established perturbed processes in the field of visualizations including those that take into account the human factor (the sketches of the formalization in the form of a discussion are given).

Keywords: signal graphs (graph signals), dynamic systems, load balancing, entropy, visualization of a digital surface model.

1. Introduction

There is such a theoretical direction as software verification. Since visualization is technically also a calculation we would like to have a common mathematical verification model for both software and visualization. Visualization verification is a formal (mathematical) proof that the visualization is correct. The developed mathematical models should answer the question whether we have correctly solved the problem set by the user, evaluate the quality and effectiveness of visualization and evaluate the prospects for the development of modern trends in the field of visualization. In particular virtual reality (VR), web-based visualization, online visualization, big data and parallel computing.

For verification, stochastic differential equations (SDE) are chosen as the basic model, as the most general solution that takes into account noise, in combination with information theory. This hypothesis is called stochastic semantics.

The initial model of image processing is the noise reduction model, which has become widespread, in particular, due to the proliferation of packages such as “Stable Diffusion“, which generate an image from text. Three variants of noise reduction are described [1]:

1. Denoising Diffusion Probabilistic Models (DDPM).
2. Noise Conditioned Score Networks (NCSNs).
3. Stochastic Differential Equations (SDEs).

As noted, the first two models are a special case of the diffusion process, which can probably be obtained using the Markov property, which is not enough from the point of view of verification. In principle, other works on this topic also use an engineering approach, that is, they show with examples that the model is working, and validation is performed with insufficient verification (plausibility is usually used). The most widely used approach is estimated networks with conditional noise. In the same paper [1], it is said that it is used to solve the transport problem namely to generate a superresolution. This approach is also known as baking. You can also refer for example to baking normal in Blender.

In principle, the same approaches are used for speech recognition in this case, the spectrum consists of phonemes in terms of visualization the spectrum is RGB.

We can recommend to study the book [2] on image processing on signal graphs (a special case of SDE) with sufficiently transparent mathematics. A signal graph is a network whose nodes contain function values. Therefore, you can define differentiation operators in the nodes and edges of the graph and proceed to partial differential SDE. Initially, signal graphs were used in electrical engineering where the approaches under consideration are a problem of automatic control the purpose of which is to construct a transfer function, for example, according to Mason's formula. The diffusion method is also used for load balancing [3].

At the same time, i would like to see a certain pragmatism in addition to theory, for example, the use of modeling to solve specific problems. In this paper, two problems are presented one in the field of parallel computing the other in the field of visualization: formalization of a dynamic system (DS) of the online visualization and parallel computing on signal graphs, visualization of grids as a parameterized model of a white-noise random process.

2. Stochastic semantics

Let us consider the visualization (interactive process and animation) from the point of view of dynamic systems. We define a visual process simultaneously as a parallel process considered as interacting sequential processes (Hoare’s processes) from the programming point of view and as a random process from the point of view of mathematical modeling. A random process is a parameterized set (family) of random variables [4]. However, unlike typical SDE models two linearly independent parameters are introduced: the amount of data and time (two random variables) - $\omega = \omega_{\Omega n} \times \omega_t$, where $\times$ is the Cartesian product or currying depending on the model. Information theory is needed to move from the product to the sum using the additivity property entropy. That is we define a random process as a composition of functions:

$$f(t, N, \omega) = f_t(t, \omega_t) \circ f_N(N, \omega_N)$$

Moreover, the amount (volume) of data N can be arbitrarily large, for example, it is not included in the operational memory of a single processor.

In general, it is necessary to consider the sum of such compositions, for example, the composition for two types of random events such as messaging and reactive computing. We note some differences between the general case and the Kolmogorov-Arnold theorem, which states that every multidimensional continuous function can be represented as a superposition of continuous functions of one variable:

1. The paper uses a probabilistic approach (random processes are considered) it is probably possible (it is necessary in the future) to switch from SDE and information theory to weak KAM [5] and mean field theory.

2. The function of many variables is enhanced by the spectral property in fact it is vector functions of many variables.

3. Data filtering is considered as a special case of reactive computing. The order of filter execution is important for the filter pipeline.

4. Unlike the amount of information, the assessment of information quality is subjective. An important task is to formally define concepts such as context, quality, cognitiveness and perceptivity of information in a way that reduces the level of subjectivity, so that can use SDE or weak KAM to solve tasks. In the field of visualization, the solution of the inverse problem is often considered (in category theory, the concept of the inverse limit is introduced), for example, one can propose such a type of display that it is continuous from the point of view of visual perception (perceptual continuity).

For software verification, as well as for visualization, along with denotational semantics, event structures can also be used by SDE. Stochastic semantics can be defined as game semantics with noise, or you can immediately use SDE. Although these two approaches are equivalent and the data flow model is considered in relation to programming they result in different mathematical models.

Verification is based on the transition from declarative definitions to formal (verifiable) definitions. In [6], the semiotic definition of the visualization metaphor is considered as a continuous mapping of the source set to the target set. At the same time, only the continuity property is added to the standard definition of a metaphor according to Lakoff [7]. A similar and most well-known approach in the field of software verification is Scott's denotational semantics [8]. Because of applying an axiomatic (semiotic) approach to formalization of visualization and parallel data filtering some analog of the stochastic control problem with morphological uncertainty is obtained. The result obtained is more mental (useful for understanding the nature of the phenomenon) than mathematical. In this paper, we consider problems within the framework of a priori uncertainty, namely SDE and information theory.

Consideration of problems related to really big data inevitably leads to the use of a block approach, which is also used in information theory and stochastic differential equations. Big data is the extreme (at the moment) case of data processing, in which universal approaches to analysis and visualization do not work or are ineffective. Then multidimensional and multicategorical data, large-volume data, or data with incomplete information (a model with uncertainty) can be considered as big data. The limit case forms the challenges that need to be answered in order to move on. Solving emerging problems leads to the fact that today's "big data" becomes the norm tomorrow [9]. When analyzing and visualizing big data, the consideration of marginal uncertainty it is the uncertainty that has a finite limit in a particular metrized topology is forced. Since computable models are being developed, we will explain this definition using the example of a computable function, which uses a special element in its definition, meaning the uncertainty corresponding to the case when the algorithm hangs. Since the process is considered, the algorithm cannot hang it is always interactively resolved as a result of debugging the correctness and efficiency of the program. Why hasn't the asymptotic theory of algorithms or automata predicted by R. L. Stratonovich been proposed so far?

3. Parallel computing

Parallel computing is a generator of more data. In some cases, instead of storing all the data it is preferable to process it using visual analytics and then decide whether this data is needed. As a result, there is a need to implement online visualization. If you need to store data, then it should be placed not on the client, but in the cloud or on a parallel file system, as a result, there is a need to implement remote visualization. Online and remote visualization tasks are architecturally very similar, so there should be a single development environment for such programs. As already mentioned, visualization is the same as computing so a unified online visualization and parallel computing environment is being developed. In [10], the main attention is paid to the validation of this environment, including the problem of load balancing by means of visual analytics. In this case, we will focus on verification, namely, the formalization of a dynamic system of online visualization and parallel computing. The main attention is paid to evaluating the efficiency of parallel programs, which takes into account the amount of data, namely, the problems of maximizing entropy for block models, where the block processor has physical and logical levels.

Before proceeding to formalization, let's list the main specifications of the online visualization and parallel computing environments:

1. Versatility of the system-online visualization and parallel computing;
2. Consideration from the point of view of dynamic systems.
3. Applicability of parametric (parameterized) models, including control via parameters.

First, visualization is technically a computation (of course, with some special features), so we can offer a universal programming method, including a programming language that will allow you to effectively implement visualization algorithms, their connection with parallel computing programs, and these programs themselves, that is, arbitrary computational algorithms.

Secondly, this environment is built around the idea of presenting a parallel computing process in the form of a set of dependent tasks [11]. Task dependency means that the input arguments for some task are determined by the results of calculating other tasks. The list of tasks and dependencies between them is determined by a custom algorithm, which can take either the form of a final calculation or a random process. Delays that occur before receiving and transmitting data will be considered as noise, which ideally tends to zero. Tasks are sent as input to the scheduling process, which determines on which computing node a particular task should be performed. Then, as dependencies are resolved, tasks are executed. Naturally, the efficiency of the entire parallel computation depends on the quality of the planning algorithm used. The assignment of a task to a node is called an assignment. The set of such assignments and dependencies between tasks will be called an execution plan. We emphasize that the execution plan is built over time, as tasks and their arguments (data) are received. As a result, a load balancing formula was proposed experimentally in [10]. One of the tasks of this paper is to show that the formula corresponds to the problem of interpolation on signal graphs [2]. For which the function value should be defined in nodes. This function can be considered time-dependent, but the real value of the data transfer time can only be obtained after a certain step of program execution, i.e. offline, which is unacceptable. Therefore, in this case, we consider a function that depends on the amount of data (similar to computational complexity). Indeed, the complexity of a parallel algorithm issometimes defined as the ratio of the complexity of data transmission to the complexity of a sequential algorithm, for example, as $O(N^2)/O(N^3)$. Of course, this is a rough expert assessment. The question arises: is it possible to refine this estimate, for example, by defining the transfer function in terms of a canonical decomposition, for example, Taylor?

Third, the most important feature of the online visualization and parallel computing environment is the ability to control both calculations and visualization through parameters. For example, parallel rendering can be organized not as a loop that depends on the camera position, but as a reactive calculation. Such approaches not only save memory, but are also in demand for research tasks, in addition, they are formalized using CDS. As an example, we will consider visualization of grids as a parameterized model of a white-noise random processor. In particular, for visualizing a digital surface model.

Let's move on to formalizing the dynamic system of online visualization and parallel computing. In addition to the formal description of this automated programming system, as already noted, we are interested in two tasks: automatic load balancing and evaluating the efficiency of parallel computing, taking into account the amount of data.

3.1. Formal description of a dynamic online visualization and parallel computing system

Let's consider the construction of a universal algebra, which we will call the process structure: $P\mathfrak{U}(\mathbb{N}, +, \circ, t\{\omega_i\})$, where $\mathbb{N}$ is the set of natural numbers which displays object identifiers, $t\{\omega_i\}$ is the logical time that depends on a list of random events.

We define the goal of automatic load balancing as follows: for any *pu process,* construct a binary homomorphic mapping from the process structure G is a signal graph that maps the process to CO-OPN (Concurrent Object-Oriented Petri Nets [12]) with minimal noise (delays): $G: pu \to CO - OPN$. Further analysis showed that Polish spaces are considered.

A Polish space is a space that is homeomorphic to a complete metric space with a countably dense subset. In particular, in the field of visualization, we consider an example for a separable Banach space. Thus, there is a possibility to switch to weak KAM.

CO-OPN, a parallel object-oriented Petri net – is one of the most well-known implementations of the algebraic Petri net. A petri net is a discrete dynamical system (directed bipartite graph) specially designed for parallel computing. Since any tree, including a syntax tree can be represented as a bipartite directed graph, the homomorphic image of this map G cannot be anything other than modifications of Petri nets. The motivation for using signal graphs is to obtain an autonomous system of differential equations. A similar approach to the use of signal graphs is graph-symbolic programming [13]. (You can also cite other examples of graph programming automation, for example, information graphs. Probably the list of works on this topic is extensive, but the authors focus not on programming, but on formalization, in particular on the development of computational complexity theory for parallel computations). In our opinion, the introduction of the arc type (sequential, parallel, terminating) is superfluous, since there is a concept of a graph path. From the point of view of programming automation and in order to avoid unnecessary copying of data, and possibly getting into the cache, the rule of maximizing the path of the signal graph (task graph) on each processor should be fulfilled, which can be represented as a function of price. In the Petri net, there is a concept of transition, similarly, in graph-symbolic programming, the predicate label is used, and in the dynamic system of online visualization and parallel computing, the standard language constructs futures (promise) and mixins are implemented at the lower level, but in a parallel version. A promise is a predicate that depends on the data type (object parameters). In this case, there is a direct analogy with colored functional Petri nets. An impurity is a predicate that depends on the processor number, and therefore depends on the parallelization scheme. In the context of big data, impurity implementation is a significant addition to Petri nets. Further, we will show that from the point of view of information theory, an effective implementation of admixtures should be based on broadcasting and vector routing of streams, in particular on message buffering. Some examples of impurities will also be given.

The logical time determines the order of events. Two types of events are possible in the DS: reactive computing and message exchange. Since entropy is additive, from the point of view of information theory, these two problems can be considered independently, for example, you can enter two logical times. Although the authors are interested in formalizing tasks related to messaging, we will focus a little on reactive computing.

In reactive computing, a random event is a change in the value of a parameter (online or offline). By analogy with abstract data types, this approach is called parameter abstraction. The simplest implementation of online visualization, but probably not the most effective, is offline parameter change. That is, all the data of the current iteration are on the client, while the user is engaged in visual analytics, the next iteration can be calculated on the computer. In this case, there is no problem with establishing the order of events, so two orthogonal logical times can be introduced. Formally, the abstraction of parameters can be considered as a lambda application and, for example, the apparatus of category theory can be used. Formally, parameter abstraction can be considered as a lambda application and can be applied to the apparatus of category theory. Although a monad is defined as a functor with an additional structure in the context of big data, there are fundamental differences between the two. For functors, there is no problem of asymptotic convergence; it is sufficient that the parameters are linearly independent or that the multiplicativity property is satisfied (in category theory the term currying is used). For monads, convergence can be considered if a repeated integral through a multiple integral is defined, for example, for lattices, space – filling curves, tessellations, random graphs, otherwise reducing a syntax tree or graph of tasks is a discrete optimization problem on graphs. In fact, in DS, a promise is a monad. Thus, the structure of processes is not just a construction of a universal algebra, but an algebraic system. a signature is a set of functional and predicate symbols with their arities.

Problems related to the exchange of messages will be considered from the position of a priori uncertainty, although the authors assume the answer. So evaluating the effectiveness of parallel computing - This is an inhomogeneous Markov control problem, where the right-hand side of the equation is equal to the number of blocks (processors) in the block model. Therefore, the problem of automatic load balancing (scheduling algorithm) corresponds to the problem of interpolation on signal graphs. A scheduling algorithm is an algorithm that selects which executor should perform a particular task. Since the amount of RAM on different classes of processors is different (and perhaps заранее и not known in advance), the transport problem can be considered as a solution, for example, with a variable number of outputs. It seems that there is a solution to this statement, like the problem of automatic control, it is constructing a transfer function using the Hurwitz polynomial, but the DS uses a different approach: if there is not enough memory, the task is transferred to another executor as a result, load unbalancing is possible. For the purpose of generalizing modeling we will refer to the executor as a processor.

3.2. Automatic load balancing task

Let's consider the problem of interpolation on signal graphs. To get started in the visualization area, see Figure 1.

Let be a function $f^0: \mathcal{V}_0 \to \mathbb{R}$, where $\mathcal{V}_0 \subset \mathcal{V}$ is a subset of the vertices of the graph with known values. The interpolation problem is reduced to considering the equation: $\Delta f(v_i) = 0$ on $\mathcal{V} \setminus \mathcal{V}_0$. In the original source [2], the Laplacian is defined in a specific way for isotropic and anisotropic diffusion processes (in fact, the Nabla operator is redefined), which is not essential for further reasoning. Simple examples of parallelization such as isotropic diffusion processes are considered in this paper. The authors believe that such examples of parallelization as discrete optimization problems on graphs and the solution of system of linear algebraic equations by the Cholesky method are anisotropic diffusion processes.

Figure 1: Interpolation scheme [2]

The algorithm is designed to dynamically distribute the task graph among processors, so that they are not idle. For the problem [14] of parallelizing a one-dimensional array on a line of processors with a shadow face of one width, that is at each iteration step the function was calculated $f(mu[i-1], mu[i], mu[i+1], \sigma)$, σ is the average value of the function at the previous step. Experimentally, in combination with visual analytics, the following load balancing formula was obtained [10]:

$$E_i = \left|\mathcal{V}_i \backslash \mathcal{V}_i^0\right| + k\ ln(queue_i),$$

where the mathematical expectation of the processor load is the lowest value of the processor load E_i (the next task is assigned to the executor with the lowest value E_i), $\left|\mathcal{V}_i \backslash \mathcal{V}_i^0\right|$ is the number of task arguments, missing from the executor cache, $queue_i$ is the current size of the list of assigned and still unresolved tasks (message queues). The computational complexity of pyramid (optimal) queue processing has a logarithmic relationship. One of the problems was the selection of the coefficient $k = 0.1$. It can be assumed that it is a characteristic of the parallelization architecture and is equal to the ratio of the (physical) channel bandwidth to the processor's computing speed.

To get to the general case, let's consider another scheme of geometric parallelization: a uniform distribution of data (a two-dimensional array) across processors is given, $N \times N$ is the amount of data on each processor, N is a variable parameter, $f_i = N^2$, ∇f_i=2N . The interpolation problem for load balancing is written as follows:

$$\nabla E_i = \nabla\left|2N(\mathcal{V}_i \backslash \mathcal{V}_i^0)\right| + \nabla k ln(queue_i) = 0.$$

Obviously, for the example under consideration, 2n is the length of the message. In accordance with the Markov property of diffusion processes [4] $\lim_{i\to\infty} \nabla k ln(queue_i) = 0$ (white noise is considered, the general name is the Poisson problem, and the other term corresponds to the Dirichlet problem).

A stochastic process has the Markov property if the conditional probability distribution of future states of the process depends only on the current state, and not on the sequence of events that preceded it.

Instead of a mathematical formulation of the Markov property, a reference to the following example is sufficient.

Example 7.3.4. Brownian n-dimensional motion is, of course, the solution of a stochastic differential equation: $dX_t = dB_t$.

Thus, the generating operator A of the process B_t (Brownian n-dimensional motion) has the form:

$$Af = \frac{1}{2}\Delta f.$$

For the general case, the following load balancing formula is quite plausible:

$$E_i = \left|d * \nabla f_i(\mathcal{V}_i \backslash \mathcal{V}_i^0)\right| + k ln(queue_i),$$

where d is the width of the shadow face, $\mathcal{V}_i \backslash \mathcal{V}_i^0$ is the number of expected messages, not the number of task arguments that are missing from the executor's cache, since message buffering must be used for optimization purposes.

The current direction in programming is vector flow routing. We will briefly focus only on message buffering. In practice, you need to consider the average message length in the number of buffers (the useful buffer size is a constant). Obviously, buffering short messages significantly increases the speed of exchanges, and for long messages it is no worse, therefore, for messages of arbitrary length, it is also advantageous to use buffering. Since synchronization commands are characterized by high latency of short messages, and buffering is not always possible, there is a desire to abandon synchronization, and nondeterministic messages will be considered further.

Although the load balancing formula looks quite plausible from the point of view of an expert programmer, the authors promised to consider the tasks set within the framework of a priori uncertainty, namely, SDE and information theory.

3.3. Information theories and the messaging model

With any formalization, a certain idealization is inevitable. Although the authors focus on the messaging model, this model is also applicable for other parallel architectures: for shared memory and accelerators, with minor additions. Modeling is based on the application of R. L.

Stratonovich's monograph: "Information Theory" [15], which fundamentally rejected the use of special terms of information theory in order to generalize it and thermodynamics. The authors, on the contrary, intend to explain some formulas in terms of information theory and CDE. R. L .Stratonovich considered the encoding (transmission) of information, the authors consider the transmission (exchange) of messages, which is basically the same thing.

The maximum value of entropy is called the bandwidth (information capacity) of a channel (computer, parallel program) without interference. Consider the problem of maximizing entropy (performance) while maximizing the path of a signal graph on each processor, which, indeed, resembles the formulation of the transport problem of fractional-linear programming: maximizing matchings while maximizing flow.

Without prejudice to the theory, the concept of "messages" can be replaced by the concept of "random variable", the concept of" "message sequence" by "random process". So the amount of information in the context of probability theory is represented as the average entropy:

$$I = H_{\xi} = -\sum_{\xi} P(\xi) ln P(\xi),$$

where ξ is a discrete random variable, and $P(\xi)$ is its probability distribution.

This formula is a consequence (in an asymptotic senseсле) of the Hartley formula, for non-probable events, which is represented as a random entropy (that is, entropy is a random variable) as:

$$H(\xi) = -lnP(\xi),$$

with the normalization condition: $\sum_{\xi} P(\xi) = 1$.

Naturally, the average entropy is the average value of random entropies:

$$H_{\xi} = MH(\xi).$$

The authors would prefer to present information theory immediately in terms of SDE. So, in the case of a continuous random variable, is it possible to use the Ito integral instead of the sum, since it is known that the differential entropy is unlimitedн? And the solution is also known: we must consider the normalized entropy, which in the context of fuzzy sets is called improbability entropy, or: we must consider the Radon-Nikodim derivative.

When defining the structure of processes, the property of additivity of entropy was mentioned earlier.

Theorem 1.3. If the random variables ξ_1, ξ_2 are independent, then the total (joint) entropy H_{ξ_1,ξ_2} decomposes into the sum of the entropies:

$$H_{\xi_1,\xi_2} = H_{\xi_1,} + H_{\xi_2}.$$

Theorem 1.4. Entropy has the property of hierarchical additivity:

$$H_{\xi_1,\dots,\xi_n} = H_{\xi_1} + H_{\xi_2|\xi_1} + \dots + H_{\xi_n|\xi_1,\dots,\xi_{n-1}},$$

where $H_{\xi_2|\xi_1}$ – conditional entropy.

This property is used in practical terms when implementing mixins. Consider a problem for which we need to calculate the mathematical expectation Mf_i, where f_i is the average value of the function on the i-processor. Based on the sequential option, in order to increase efficiency, we can offer pairwise pyramid summation, but in this case the messages will be point-to-point, that is, non-probable, and the channel is not symmetrical. Perhaps for a small number of processors, this option will be more efficient than the implementation taking into account the hierarchical additivity property: Using broadcasting, summation should be performed on each processor, while the selection tree on each processor will be different, but the conditional entropy on each processor will still tend to zero. This parallelization scheme is similar to the master-worker scheme without synchronization, for which each processor is both a master and a worker.

To implement a selection tree, each processor must have at least two (physical) bidirectional communication channels, where k is the number of processor channels similar to the number of letters of the alphabet (for a multi – core architecture, k is the number of cores, but the selection tree is directed in the other direction, similar to pyramid summation). In addition, there is a not entirely correct opinion that the result of summation depends on the order of summation. In the case of entropy stability, all implementations of summation will be approximately equal.

Amdahl's law illustrates the limitation of the performance growth of a computer system with an increase in the number of computers (processors), which is formulated as a well-studied Bernoulli distribution. That is, Amdahl's law is a special case from the point of view of information theory and does not take into account the exchange of messages. Due to the property of additivity of entropy, these two problems can be considered independently. In addition, for geometric parallelization with a fixed amount of data, the share of sequential calculations is inversely proportional to the number of processors, that is, it tends to zero with the number of processors tending to infinity, therefore, when evaluating performance, only the exchange of messages should be taken into account. When considering the general case, we need not only the equality of processors, but also the equal probability of messages (asynchronous non-deterministic messages, stationary process by the number of processors). Since the cluster architecture is most widely used, in which exchanges are carried out via a common bus (we can also consider a k-tree), such an idealization is justified, in addition, the channels are two-way, so we can assume that the path length for asynchronous messages is one, and for synchronized exchanges it is two. The path length is a constant that depends on the message type and the architecture of the computer, which must be multiplied by the average message length.

In fact, equal probability of messages is not required. You can consider deterministic messages (a steady-state process), for example, the data grid on the processor grid, but this is a rather narrow class of problems. anisotropic diffusion processes and multiple integrals are considered as an extension of the class of problems.

The concept of entropic stability of a family of random variables helps to give a general formulation of the property of asymptotic equivalence of non-equiprobable possibilities (messages) to equally probable ones.

A family of random variables $\{\eta^n\}$ is entropically stable if the ratio $H(\eta^n)/H_{\eta^n}$ при $n \to \infty$ converges to unity in probability. This means that whatever they may be $\varepsilon > 0$, $n > 0$ there is $N(\varepsilon, \eta)$ such that the inequality holds:

$$P\{|\,H(\eta^n)/H_{\eta^n} - 1| \geq \varepsilon\} < \eta \ ,$$

for any $n > N(\varepsilon, \eta)$.

The definition implies that all $0 < H_{\eta^n} < \infty$ and H_{η^n} does not decrease with increasing n. Usually $H_{n^n} \to \infty$.

The fact of asymptotic equiprobability can be formulated using the concept of entropic stability in the form of the following theorem.

Theorem 1.9. If a family of random variables $\{\eta^n\}$ is entropically stable, then the set of realizations of each random variable can be divided into two subsets A_n and B_n in such a way that

1. The total probability of the subset A_n vanishes:
$$P(A_n) \to 0 \text{ for } n \to \infty.$$
2. Realizations of the second subset B_n become relatively equiprobable in the sense of the relation
$$\left|\frac{lnP(\eta) - inP(\eta')}{lnP(\eta)}\right| \to 0$$

for $n \to \infty, \eta \in B_n, \eta' \in B_n$.

3. The number M_n of realizations of the set B_n is related to the entropy H_{η^n} nn by the relation
$$lnM_n/H_{n^n} \to 1 \text{ for } n \to \infty.$$

Here are some comments on entropic stability. From the point of view of SDE, the definition of entropic stability can be considered as a generalization of the Ito integral through the probability limit: The ratio $H(\eta^n)/H_{\eta^n}$ is the relative entropy or Kullback-Leibler distance for a uniform distribution. In this case, the notation $H_{\xi}^{P/Q}$.

In information theory, the main focus is on current coding but another approach, which can be called block-based, is also considered. In this case, the final set (block) of elementary messages must be encoded. If the block is an entropy-stable value, then the probability of losing some of the message implementations is quite small. As already noted, instead of encoding messages, you can consider message transmission, and the maximum value of entropy is called the bandwidth of the channel without interference. Let us consider the problem of maximizing entropy in the case of the block approach. As a comment, we note that this problemтна задаче is equivalent to the Kullback-Leibler distance minimization problem Кульбака-Лейблера, also called the maximum likelihood problem. In our interpretation, a block is a processor that must be entropically stable, that is, a PROCESSOR=MARKOV PROCESS, taking into account hardware and software implementations, as well as the entire computer. If we consider the problem of fault tolerance for a Markov process, then one additional processor is sufficient for its implementation (the average probability of failure of one processor is very small), and events associated with replacing one processor with another do not affect the performance of calculations (without taking into account data saving and recovery).

The part of the task that is implemented on the processor (block) must be a Markov process.

Here are some examples of block Markov processes.

The authors constantly refer to geometric parallelization, since in this case the interpretation is quite clear. The data located in the block is an open set, since there are shadow faces. The total measure on the boundary of the set must be less than the measure of the interior of the set. In fact, this is a variant of the Chebyshev inequality, which is actively used in proving theorems.

In principle, we can consider a graph (of tasks on the processor) of a rather arbitrary structure. Without diminishing generality, we can consider the Radon-Nikodim derivative by introducing two measures, and, consequently, a random process:

$$E \le X \to \frac{1}{X} \le \frac{1}{E(X)} \ll 1,$$

where X is a measure defined in the nodes of the graph and depends on their number, *and* E is a measure defined on the arcs or border of the graph.

These relations are also valid for signal graphs, for example, the number of nearest neighboring vertices. It is also applicable in other areas of knowledge, for example, for graph visualization, in particular, knowledge anthologies. “For a hypergraph h with a given set of nodes X, a given set of edges E, and a set of corresponding values of the incidence matrix $\{a\}$, the information I(h) has the following form [16]:”

$$I(h) = \frac{1}{|X|} log_{|\{a\}|}|E|.$$

Although the incidence matrix is used to reduce graphs, including Petri nets, in the authors 'opinion, the use of the determinant of the incidence matrix in the base of the logarithm is not justified.

Let us describe the Kullback-Leibler distance minimization problem:

$$\lim_{X\to\infty} D_{KL}\left(\frac{1}{X}\middle|\frac{1}{E(X)}\right) = \sum \lim_{X\to\infty} \frac{1}{X} \ln\left(\frac{E(X)}{X}\right)$$
$$= \sum \lim_{X\to\infty} \frac{1}{X} ln(E(X)) - \sum \lim_{X\to\infty} \frac{1}{X} ln(X) = \sum (I(h) - 0)$$

In fact*, I(h)* is the relative entropy, and the conditional entropy tends to zero.

A well-known example is the case when the choice tree is bounded from above by a k-tree, which in information theory is called optimal encoding of decipherable Kraft codes.

Theorem 2.3. It is possible to specify such a method of encoding (transmitting) equidistributed independent messages that

$$l_{cp} < \frac{H_\xi}{lnD} + 1.$$

In our interpretation, l_{cp} is the average path length between computer nodes, and D is the number of processor channels (in the original, the number of letters in the alphabet). In the case of message exchange via a shared bus, we can assume that $D = \infty$, therefore, the average path length betweenthe computer nodes is equal to one, which corresponds to the expert approach. In the case of the block approach, the following theorem applies.

Theorem 2.4. There are ways to encode an infinite message such that the average length of an elementary message can be made arbitrarily close to

$$\frac{H_\xi}{lnD}.$$

Such estimates for the average path length are applicable not only for the hardware architecture of the computer, but also at the logical level for graphs in the space - R^D. Useful references are out-of-core algorithms (k-tree data restructuring) [17] and "Polynomial approximate scheme for the problem of cyclic covering of a graph of fixed size k in a Euclidean space of arbitrary fixed dimension" (discrete optimization) [18].

Since the processor does not exchange messages with itself, you can introduce the concept of subentropy, for example, for equally probable messages, similar to the definition of subfactorial. Subfactorial $!n$ is defined as the number of disorders of order n, that is, permutations *of an* n-element set without fixed points. In the case of asymptotic convergence, the equiprobability of messages is not mandatory, as is the consideration of subentropy. In [19], it is shown that in classical information theory, the subentropy dual of the von Neumann subentropy, defined through permutations of the pairwise eigenvalue difference, is an exact lower bound on the channel throughput and its calculation corresponds to the Kullback-Leibler distance minimization problem.

Depending on the task, the message length between different processors may not be the same. In this case, the main characteristic is the average message length. Let us consider a direct method for calculating the maximum entropy for this example, which corresponds to section 3.1. Let there m processors - $V_1, \ldots, V_m$ that transmit messages of length $l(1), \ldots, l(m)$ respectively. The total message length will be $L = l(1) + \cdots + l(m) = l_{\text{cp}}m$. We fix this length and count the number $M(L)$ of different implementations of this length. Maximum information is obtained when all of the $M(L)$ possibilities are equally probable. At the same time

$$\frac{H_L}{L} = \frac{lnM(L)}{L}.$$

Taking the limit for $L \to \infty$ we obtain the entropy calculated per unit length. That is, it suffices to consider the solution in the case of asymptotic convergence of a linear homogeneous equation which has the form - $M(L) = Ce^{\lambda L}$. The solution has a unique root with maximal real part λ_m. Andso, the number of different implementations of length L has the form:

$$M(L) \approx C_m e^{\lambda_m L}.$$

The same result can be obtained from solving the first variational problem. We restrict ourselves to only considering a discrete channel without interference (a general statement of the problem). The system $[Y, c(y), \alpha]$ completely characterizes the discrete channel without interference where $y \in Y$, the penalty function is $c(y) \leq \alpha$ (microcanonical distribution). In particular, $c(y) \leq L$. It is more convenient to consider the canonical distribution - $\sum c(y)P(y) \leq \alpha$. The bandwidth C or information capacity of the channel $[Y, c(y), \alpha]$ is defined as the maximum value of entropy - $C = \sup_{P(y)} H_y$. Thus, the channel capacity is defined as the solution of variational problemsand. Note that it is equivalent to the problem of minimizing risks (time delays). The equivalence of the microcanonical and canonical distributions is proved. In the context of this paper, Amdahl's law is a microcanonical distribution, and the corresponding canonical distribution is the Bernoulli distribution. We also emphasize that the problem of maximizing the performance (entropy) of parallel computing is considered.

To illustrate the importance of the average message length, here is an example of the performance behavior of a parallel program with a fixed amount of data and a variable number of processors. As the number of processors increases, the average message length in the number of buffers may decrease by one and as a result of the exponential dependence on the message length, there should be a jump in performance. Thus, the message length may depend, as well as the percentage of consecutive computations, on the number of processors (a variable parameter essentially similar to time). Therefore, we need to consider the solution of a linear inhomogeneous equation, and first write out the penalty function.

3.4. Optimal scalability of parallel computing (block approach)

In the context of high-performance computing, there are two indicators of scalability:

1. Strong scalability-shows how the time to solve a problem changes with an increase in the number of processors (or computing nodes) while the total task volume remains unchanged.
2. Weak scalability-shows how the time to solve a problem changes with an increase in the number of processors (nodes) while the task size for one processor (or node) remains unchanged.

Optimal scalability (an estimate of parallel computing performance that takes into account both the amount of data and the number of processors) is a non-uniform Markov control problem, where the right-hand side of the equation is equal to the number of blocks (processors) in the block model. Of course, this is another plausible hypothesis.

By optimal scalability, we will understand the case when the channel throughput is defined as the solution of a variational problem, that is, the entropy can be written out explicitly. First, you need to define the penalty function (for a discrete channel), which depends on two parameters: the amount of data and the number of processors – p:

$$c(N,p) = s\sum_{i=1}^{p} l_i(N,p) + \sum_{i=1}^{p} \alpha_i(N,p),$$

where α_i is the share of consecutive calculations of the total processor, s is the average path length (recall that for equally probable messages, it is equal to one). In addition, the use of signal graphs was previously assumed as a development of ideas of computational complexity that is - $l_i(N,p) = \nabla f_i(N,p)$, which will lead to the consideration of the second variational problem.

Next, we will try to find a similar penalty function in information theory. To quote paragraph 3.5 the potential method for a large number of parameters: "The penalty function depends on a numerical parameter and is differentiable with respect to this parameter." In our case, by the number of processors (blocks) is - p. The authors do not see much point in rewriting the known formulas in other notations. However, here is one definition. Function -

$$B(N) = -\frac{\partial c(N,p)}{\partial p}$$

is called a random internal (endogenous) thermodynamic parameter conjugated with the external (exogenous) parameter p. Later in the same section, we consider an example with two random variables, in our case, the message length and the proportion of consecutive calculations that are obtained from solving a system of two equations with two unknowns. Thus, the analytical solution of the problem under consideration is known.

The analytical solution can be used to determine interference in the computer. Consequently, there is a need to develop adequate models for processing statistical data on the performance of parallel computing, including for the following task. So far, only stationary processes have been considered. Using mixins, a stationary process can be implemented in the DS for the task graph with some overhead costs relative to the share of sequential calculations (obviously, the Chebyshev inequality must be fulfilled, that is the more data in a block, the lower the share of overhead costs, probably we should consider the channel with interference (paragraph 7)) and reading messages (recall that the message queue size tends to infinity, which corresponds to white noise). In the case of geometric parallelization, you can implement and compare a stationary process and a steady-state process (point-to-point messages). An analytical solution for the latter is also known in information theory (although the term steady-state process is not used and parallel computing has not been considered). To do this, we introduce the concept of communication information (in our case, graph arcs), naturally, through a conditional probability, through it A symmetric channel is also defined, naturally, using permutations, respectively paragraphs 6 and 8. In this case, the conditional distribution at the channel output for a fixed input signal is assumed to be known (in fact, the transfer function is constructed).

The channel capacity $[P(y|x), c(x)]$ is the maximum value of the communication information between the input and output:

$$\mathrm{C} = C[P(y|x), c(x)] = \sup_{P(x)} I_{x^{i,j}}.$$

In the case of a fully symmetric channel (processor grid), the formula looks quite simple (8.4.9):

$$\mathrm{C} = lnM\, {}^{1}/{}_{p_j} - Mln\, {}^{1}/{}_{p_j}.$$

In the case of a uniform distribution, the formula has the following form:

$$C = lnM \frac{Q(dy)}{P(dy)} - Mln \frac{Q(dy)}{P(dy)}.,$$

where $Q(dy)$ is an auxiliary measure based on which the intervals are rearranged.

However, this formula is performed only on the core (in the core) or in the case of shared memory, since in the cases considered [14], the penalty function depends on one variable. For geometric parallelization, you can reduce the dependencies to a single variable, taking into account that for a fixed amount of data, the share of sequential calculations is inversely proportional to the number of processors, that is, it tends to zero (but not zero) with the number of processors tending to infinity. The more data on the processor, the better the balance of calculations, which leads to a certain contradiction.

In practical terms, it is more important to follow certain programming rules when developing parallel programs, starting with a binary homomorphic map and ending with a steady-state process. If the graph of strong scalability corresponds to a logarithmic function, then the parallel program can be considered a steady-state process in terms of the number of blocks (if strictly, then we should consider the linear filtering problem [4]: systems with noise and measurements with noise). In order to increase productivity, we can consider the following extrapolation problem [6] (prediction [4]) up to the point of stopping: $N(p-1)$ is known, we need to find *N(p)* while maximizing entropy, which the authors just call the inhomogeneous Markov control problem. Since the general solution of an inhomogeneous equation is the sum of the fundamental system of solutions and the partial solution, consideration of an inhomogeneous problem automatically leads to an increase in entropy and reduces the dimension of the system of homogeneous equations (in our case, to one). An inhomogeneous problem can be obtained from random time replacement in [6], the entropy function $p/N(p)$ or fractional linearization on the kernel was considered) $L = l_{cp}p$ differentiating by the average message length. At the boundary, the condition corresponding to linear acceleration (maximization of entropy) must be satisfied - $\dot{X} = p$.

Since statistical data are processed, we need to switch to Markov chains (solving a linear system of algebraic equations), which will be the best approximation of the problem of maximizing the efficiency of parallel calculations of the message exchange model - $H_pX = p$, where H_p is the message exchange matrix (adjacency matrix). The main diagonal shows the average time of sequential block calculations while the other elements correspond to the average message transfer time between processors, it is assumed that the parallelization scheme does not change (does not depend on the iteration number).

The message exchange matrix will be close to symmetric, since a response message of the same length should presumably take a little longer. In such cases, we consider a Dirichlet problem with stochastic control of the form regulator with incomplete information about the state of the system, let's call it a stochastic information management problem. But on the other hand, this matrix will have a different symmetry associated with the load balanced calculations, a parallelization scheme, and the definition of a symmetric channel.

We can assume that the eigenvalues are responsible for the method of parallelization. It is important that the result is a matrix of certain templates, so for the master-worker scheme, the main diagonal and i-row and i-column are not zero (i is the number of the processor on which the master is located). A three-diagonal matrix is obtained for the pipeline or processor line, a five-diagonal matrix is obtained for the grid and so on.

Let's take a closer look at the master-worker scheme. The adjacency matrix has the following form:

$$H_p = \begin{vmatrix} a & b & \dots & b \\ b & a & & 0 \\ \vdots & & \ddots & \\ b & 0 & \dots & a \end{vmatrix} \begin{matrix} 1 \\ 2 \\ \vdots \\ p \end{matrix}.$$

The system is not compatible only when $a_{1,1} = 0$, time without taking into account data preparation and saving, i.e. the master does not do the calculation. From the last equation it is easy to find x_p

$$dx \approx x_p = \frac{p - bx_1}{a}.$$

In the space L_2 the entropy increment (the shift coefficient in the diffusion process) is

$$dx = \overline{X_p} - \overline{X_{p-1}} \approx x_p.$$

Taking into account the computational complexity of the algorithm $a = f(N)$ (diffusion coefficient) and $b = \nabla f(N)$, for example, $f(N) = N^3$ and load balancing by the number of processors the entropy increment has the form:

$$dx = \left(\frac{P}{N}\right)^3 p - O\left(\frac{1}{N}\right) \to \ dx \approx \frac{p^4}{N^3}.$$

It turns out that the more complex the algorithm, the better it parallelizes (scales). If we put $\frac{p^4}{N^3} = 1$ (the maximum entropy value is one), we get the expression N in *terms of* p. Of course, the above estimate is rough, but we can consider a parameterized model of a white-noise random process, since the entropy function is always $p/N(p) \ll 1$ (for more information, see the section on visualization). This function is a harmonic function (a property of the mean), therefore, the consideration of the moving boundary problem is justified, which is called the Jacobi problem in SDE [4].

To prove that the penalty function is written correctly, we can compare the graphs of the Dirichlet beta distribution function and the efficiency of parallel calculations for the problem of solving system of linear algebraic equations by the Cholesky method [20] in the case of balanced calculations, see Figure 2.

Let's $X = (X_1, \dots, X_K) \sim Dir(a)$ is Dirichlet distribution and $a_0 = \sum_{i=1}^{K} a_i$ then $E[X_{i|a}] = \frac{a_i}{a_0}$.

$$B(x, y) = \int_0^1 t^{x-1}(1 - t)^{y-1} dt$$

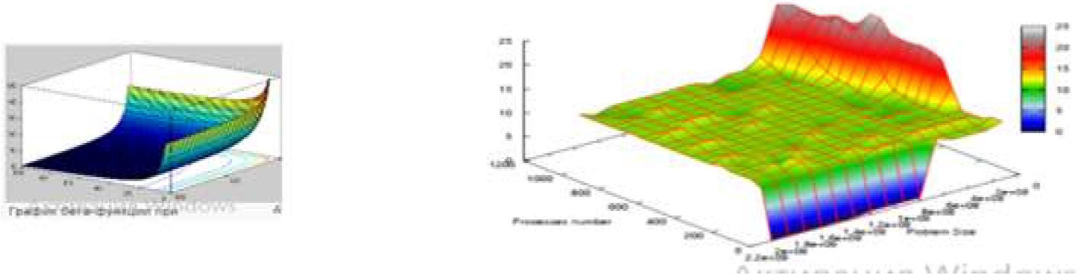

Figure 2: On the left, there is Beta function, on the right, there is efficiency of parallel computing

From the point of view of visual verification, the graphs are similar.

4. Examples of applying the block approach in other areas of knowledge

The block approach in SDE is used quite often, for example, in the global model of earth seismicity [21], but the number of blocks is fixed (for example, the number of lithospheric plates). Unfortunately, the authors could not find any examples where the number of blocks is a variable parameter. And this problem is relevant, not only for the problem of predicting the performance of parallel computing, but also for other extrapolation problems (in the general case, stochastic information management problems), which are given below.

Let us consider a similar example from the field of economics-the analysis of data from the Accounting Chamber. A matrix of deliveries by (mobile) region numbers is generated (Figure 3) and a harmonic function is defined the ratio of the sum of deliveries in the region to the number of firms in the region, its value between regions is displayed as spheres in the corresponding matrix elements. We can see that by rearranging rows from this matrix, you can get a matrix close to symmetric, taking into account the fact that they always order more than necessary (the exact upper bound is determined). Although the role of the regulator is important in the economy, it is also related to the rate of profit and the level of corruption, let's just consider whether the information gap between Moscow and other regions is narrowing. We define the information gap measure as the ratio of multidimensional distance to geographical distance, which is an auxiliary measure in the Radon-Nikodim derivative). If we consider the ratio of information gaps in the extrapolation problem, the auxiliary measure will be reduced. If the ratio measure increases, the information gap will shrink and vice versa. Now let's look at the same problem when a new region is added. Us it has already been considered, the number of blocks (regions) is a variable parameter. Typical clustering problems can also be considered, but from the point of view of dissipative systems – the formation of new clusters. The same model is used to analyze the distribution of information in Internet networks, where the parameter is not the number of regions, but the number of information channels (blogs, etc.).

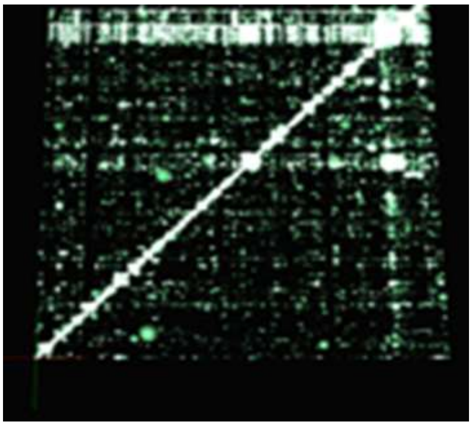

Figure 3: Delivery matrix by region

Next, we will look at some examples of stochastic visual information management problems, focusing on the application of the block approach in visualization.

5. Stochastic management of visual information

Sinkhorn (Sinkhorn neural networks based on the Sinkhorn theorem [22] are used to solve a wide class of transport problems; super-resolution, comparison of two distributions. It is argued that these networks are better in terms of speed and number of parameters than generative maximum likelihood networks. Next, we will consider the possibility of using SDS for problems on (signal) graphs as an alternative to the traditional use of neural networks. For parallel calculations, the function defined on the signal graphs depended on the amount of data in this section height dependence (height map). First, we will consider the task of visualizing a digital surface model (DSM) with a fixed number of blocks and a variable amount of data in each block. For this task, the maximum number of blocks depends on the amount of data that is limited by the video card's memory, i.e. it is a constant. In the sequel, we will consider the task of recognizing gestures of infants on one block as a comparison of two distributions (video streams) of a healthy infant and possibly with deviations in the future. In principle, for speech recognition tasks, the number of blocks can be a variable parameter, not a constant. In the future, the goal is to solve the problem of object detection as a composition of these two problems (these two random processes) with different types of heterogeneity: the number of blocks and the height.

5.1. Visualization of grids as a parameterized model of a white-noise random process

A block is a DSM data storage element (similar to the recognition problem), a matrix of size NxN, at each point of which the height function above sea level and a constant colore, code corresponding to the object class, то are defined, i.e. a signal graph on the grid is defined. The DSM is represented by a set of such matrices or a block matrix. At the same time, the block has a hierarchical structure that is a quad-tree, which reflects different levels of detail in terms of accuracy. The main specification of visualization is its application in VR (three-dimensional graphics), that is, in a Banach space (not points are displayed, but intervals) and because of this "joints" or "holes" are formed between blocks, which cause difficulties in implementing rendering algorithms. They must handle block boundaries in a special way. In Gilbert space (for raster graphics), there are no such problems. The main difference between the three-dimensional graphical approach and the standard one used in SDE is that the limit of the step function should be defined not in the pointwise convergence topology, but in a compact-open topology, which is done in order to build a continuous display (visualization) from the point of view of visual perception.

When flying around the DSM, blocks are loaded taking into account the function of the minimum distance between the camera and the block, first with the worst accuracy, and then with improved accuracy. The visualization application is implemented using Web-GL, with shader abstraction implemented, which is similar to parameter abstraction, i.e. a shader is a function that depends, for example, on the camera position. In fact, reactive calculations are implemented at the video card level. Of course, this direction is interesting, but as already noted, it will not be considered.

Here are the types of display implemented on the block. The simplest one is a point cloud, which will not be considered, since it is not a continuous display from the point of view of visual perception, which in the Laplace transform corresponds to the “original”. In Figure 4 for comparison, there are two types of display that are hardly distinguishable from each other, but with a different visualization model. On the left, standard polygonal graphics or barycentric coordinates. On the right, visualization by columns. The column is a metaphor. This type of display is sometimes called a statistical prismagram (quadrilateral prism) than is a three-dimensional analog of a diagram. From a mathematician's point of view, this is the inverse transformation for the "image" into the Laplace transform, which is what needs to be shown, first by construction. (Visualization is often considered as a solution to the inverse problem, but rather we should talk about the solution to the conjugate problem).

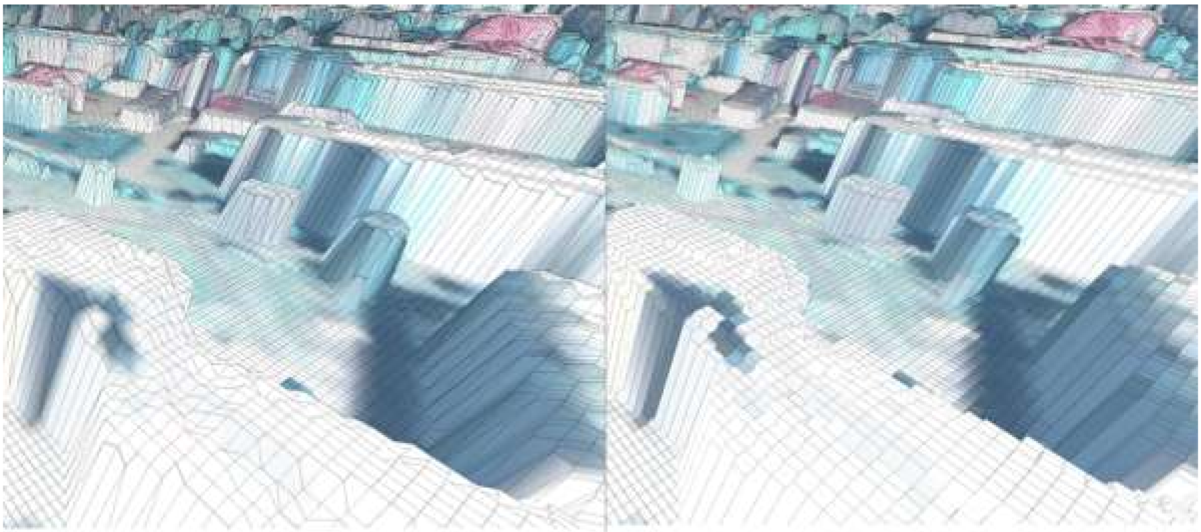

Figure 4: Tasks of visualization of a digital surface model, on the left the surface is depicted by triangles, on the right he surface is depicted by columns

Figure 5 shows how a statistical diagram is constructed for the two-dimensional case. A formal description will be given below, but for now we will limit ourselves to a declarative one using cognitive visualization. Historically, the term cognitive visualization comes from solving mathematical problems in a graphical way [23]. A statistical diagram is a directed step function, a function of height, the expected value of which corresponds to the middle of the segment, interval (shown by the red dot), which corresponds to the Stratonovich integral. In three-dimensional space, a two-dimensional interval, taking into account the detail and direction of the normals, is usually called a micro-face. For example, in computer visualization, micrograins are used to display a rough surface. The connection between the microf the facet and the cone of normals is obvious. Taking into accountthe level of detail (accuracy), we can consider a multiple Stratonovich integral with respect to spatial variables (a stationary process), which in the limit is equal to a double. In addition to the display view, “steps” are also drawn – partial derivatives (shown by the red segment), which is done for the purpose of continuity of the display from the point of view of visual perception. It is worth emphasizing that for the projection on a plane, the double integral of Ito cannot be drawn by columns, unlike the projection on a sphere or cylinder, which is planned to be used in the implementation of the wave equation of rendering as a diffusion process. It is known that for a one-way transformation, the variance tends to infinity. Of course, it would be possible to display the height value in a square, but such a drawing does not make sense. In addition, the perspective projection and affine image transformations are linear, so they do not affect the variance (for example, for the Alon dispersion), which is important when interacting directly with the DЫM.

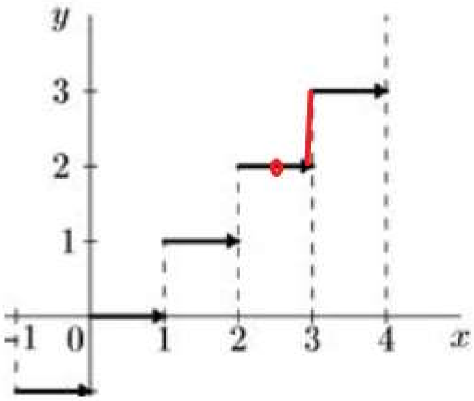

Figure 5: Statistical diagram - step function

It is worth noting that as a basic type of display, you choose visualization by columns, because, firstly, there are many vertical lines (steps) in the DSM, and secondly, the area of holes is smaller and they are located vertically, and not in a horizontal plane as in the case of polygonal graphics, see Figure 6.

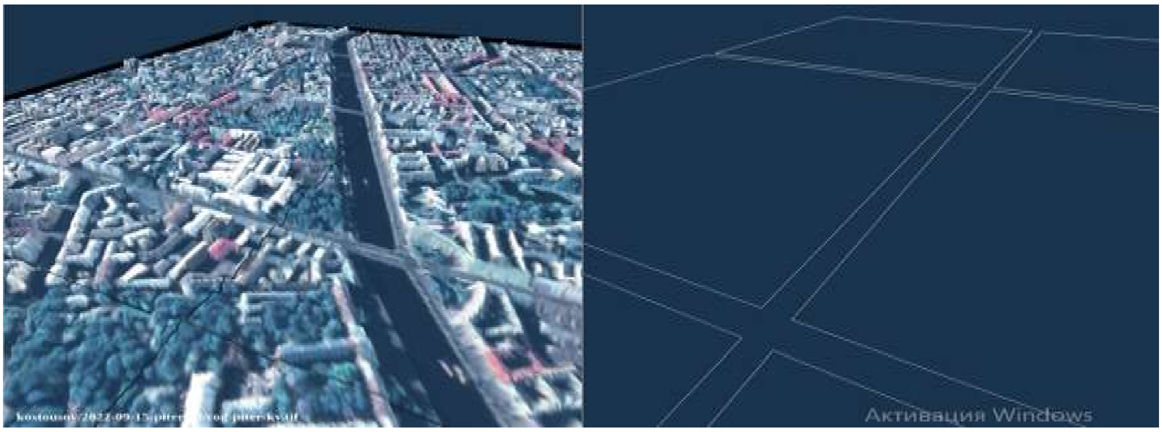

Figure 6: Artifacts are visible on the left (blue lines, background colors on the block borders), показано and the division into blocks is shown on the right

The question arises whether it is possible to remove artifacts in the image, for example, by considering the transport problem at the border of blocks. In the case of polygonal graphics, the answer is obvious: you can introduce a dummy line (shadow face) on the border, and take the average value of the height at the point (derivative). Of course, there are certain difficulties in terms of programming, which we will not dwell on. However, this approach will not work in the case of visualization by columns. Figure 7 is shown to compare the artifacts of these two types of mappings.

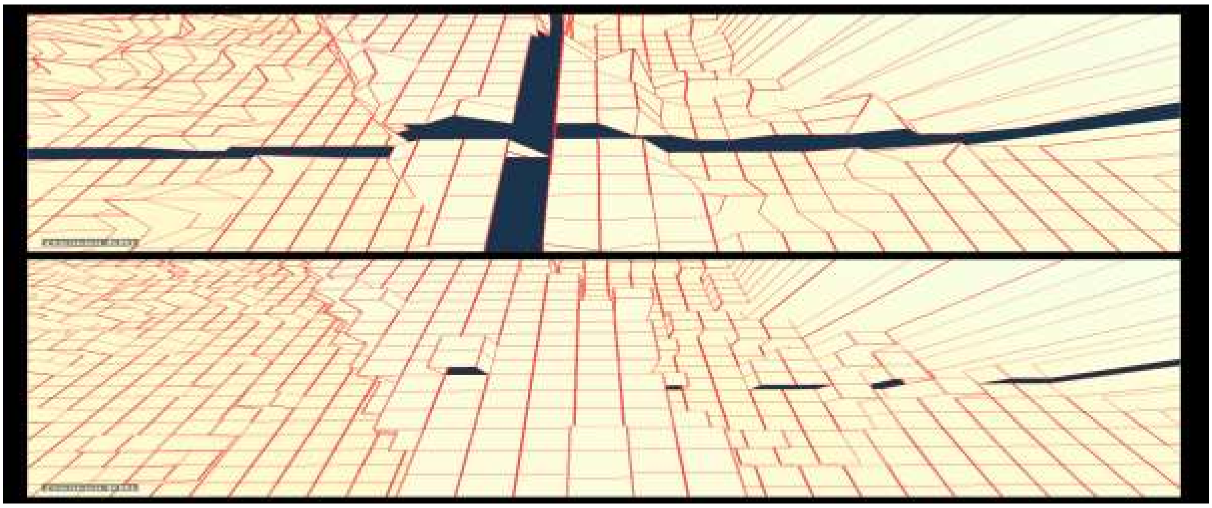

Figure 7: Artefacts at the top for polygonal graphics, at the bottom for bar visualization

What are the holes in the case of visualization by columns? This is the integral metric that is the Wasserstein metric for partial derivatives:

$$\sum_{v_i \in \overleftarrow{V_J}} ||\nabla f(v_i)||_2^2,$$

where v_i graph node, $f(v_i)$ height, $\overleftarrow{V_J}$ internal border of blocks.

Integral metric is more informative, than a vector field. It is unlikely that mathematicians have considered the problem of removing holes, probably here you can conjure with the m Dirichlet distribution or with stochastic control, but such a solution will still lead to exchanges between blocks. It is probably possible to dispense with exchanges by approximately redefining the partial derivative on the boundary symmetrically down from the previous cells, since the three sigma rule holds for the Markov inequality, which is a special case of the Chebyshev inequality.

The same approach to DSM visualization is also applicable for three-dimensional meshes, when instead of the (graphical) projection filter, a plane cross-section filter (or a sphere cross-section for the wave rendering equation) is interactively applied, as a result, it is necessary to determine the Ito formula for filters. In addition, the grid does not necessarily have to be regular, it can always be restructured by an octree. In the multidimensional case, a scattering matrix is used which is directly related to the definition of a fully specified random process. See Figure 8, where instead of the "original" (point cloud), visualization by columns is used.

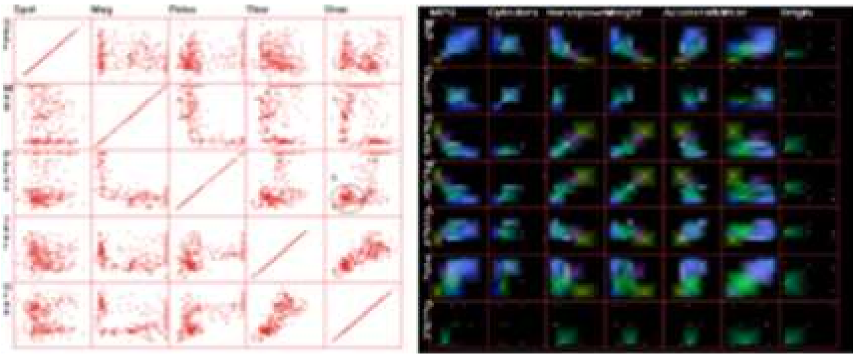

Figure 8: Scattering matrix, on the left for a point cloud [24], on the right for parallel coordinates [25], which can be considered as a complete differential [26].

This is the end of the declarative description of the problem, the formalization will mainly be based on the monograph [27] by D. F. Kuznetsov: "Some problems in the theory of numerical solution of stochastic differential equations of Ito".

5.2. Formalizing the visualization of grid task

Let's start with the introduction of the Ito integral. To get to the integral, the limit of the sum is determined in a special way using the step function [4]:

$$\sum_i f(t_i^*, \omega)\,(B_{i+1} - B_i)(\omega)$$

Ito integral $t_i^* = t_i$ is used the left end of the segment. It is denoted by:

$$\int_S^T f(t, \omega)\, dB_t(\omega).$$

The function $f(t, \omega)$ is measurable, consistent, and:

$$E(\int_S^T f(t, \omega)^2\, dt) < \infty$$

An important property of the Ito integral is that it is a martingale.

Stratonovich integral $t_i^* = (t_{i+1} - t_i)/2$ is used midpoint of the segment.

You can define **a generalization of the Ito integral** in terms of the probability limit:

$$\int_0^t f(s, \omega)\, dB_s(\omega) = \lim_{n \to \infty} \int_0^t f_n(s, \omega)\, dB_s(\omega),$$

Where $f, f_n \in W_H$, f_n is the step functions such that $\int_0^t |f_n - f|^2\, ds \to 0$ in probability (with respect to P).

It is obvious that the definition of entropic stability is also a generalization of the Ito integral through the probability limit. A similar definition is used in [26].

Definition 1.2 A sequence of random variables $\xi_k(\omega)$ is said to converge with probability one or almost certainly to the random variable $\xi(\omega)$: $\xi_k \overset{ac}{\to} \xi$ for $k \to \infty$ if

$$P\{\omega: \xi_k \to \xi\ for\ k \to \infty\} = 1.$$

This sequence is also called the fundamental convergent sequence with probability one. The fundamental nature of a sequence of random variables is a necessary and sufficient condition for the existence of its limit, which is called the Cauchy criterion.

Let us consider the difference between the standard definition of a completely specified random process and the definition in a Banach space

A random process is considered to be completely specified if its finite-dimensional distributions or a set of distribution functions are specified, which are defined for any $k \geq 1$ by the following relations:

$$F_\xi(x_1, \dots, x_k, t_1, \dots, t_k) = P\{\cap_{j=1}^{k}\{\xi(t_j, \omega) < x_j\}\},$$

where $x_j \in R^j$.

The converse statement established by Kolmogorov is also true:

If the functions $F_\xi(x_1, \dots, x_k, t_1, \dots, t_k)$ for all $k \geq 1$ satisfy the conditions:

1. $F_\xi(x_1, \dots, x_k, t_1, \dots, t_k)$ is a joint distribution function *of* k random variables.
2. The identical equality holds

$$F_\xi(x_1, \dots, x_k, t_1, \dots, t_k) \equiv F_\xi(x_{i_1}, \dots, x_{i_k} t_{i_1}, \dots, t_{i_k})$$

for any permutation $i_1, \dots, i_k$ of the numbers $1, \dots, k$.

3. $\lim\limits_{x_k \to +\infty} F_\xi(x_1, \dots, x_k, t_1, \dots, t_k) = F_\xi(x_1, \dots, x_{k-1}, t_1, \dots, t_{k-1})$

then there exists a random process $\xi(t, \omega)$ whose joint distribution functions are:

$$F_\xi(x_1, \dots, x_k, t_1, \dots, t_k).$$

Thus, the set of joint distribution functions of values of a random process $\xi(t, \omega)$ is its exhaustive characteristic.

If the distribution functions $F_\xi(x_1, \dots, x_k, t_1, \dots, t_k)$ have a finite mixed k- derivative, then there are joint distribution densities of the values of the random process ξ_t at the corresponding time points:

$$p_\xi(x_1, \dots, x_k, t_1, \dots, t_k) = \frac{\partial^k}{\partial x_1 \dots \partial x_k} F_\xi(x_1, \dots, x_k, t_1, \dots, t_k).$$

In a Banach space the matrix of partial derivatives of x, like the covariance matrix, is non-symmetric (the derivative on the left is not equal to the derivative on the right). For a symmetric matrix the number of permutations is $k!$ and for a non-symmetric matrix it is 2^k. In general, we should apply Sinkhorn's theorem [21]. A Hilbert space is a Banach space, hence we can consider symmetric matrices in the limit. In this case, multiple integrals arise. Obviously, detailing on a single block defines a compressive map, so we can apply the Banach fixed-point theorem. Gaussian processes, fundamental sequence centering and Laplace transform are typically used to solve problems numerically.

A random process is called Gaussian process if all its joint distribution densities are Gaussian:

$$p_\xi(x_1, \dots, x_k, t_1, \dots, t_k) = \frac{1}{2\pi^{\frac{k}{2}}|K|^{\frac{1}{2}}} \exp\left(-\frac{(x-m)^T K^{-1}(x-m)}{2}\right),$$

where $x = (x_{x1}, \dots, x_k)^T$, $m = (m_1, \dots, m_k)^T$, $m_i = M\{m_i\}$ and covariance matrix $K = K^T > 0$.

The process $\overset{\circ}{\xi_t} = \xi_t - M\{\xi_t\}$ is called the centered component of the process ξ_t .

The function: $R_\xi(t_1, t_2) = M\left\{\overset{\circ}{\xi_{t_1}}\overset{\circ}{\xi_{t_2}}\right\}$ is called a correlational function of the process ξ_t, where:

$$D\{\xi_t\} = R_\xi(t, t),$$

where $D\{\xi_t\}$ is dispersion of a random process ξ_t.

In order for the function $R_\xi(t)$ for $t \in (-\infty, +\infty)$ to be a correlation function of a broadly stationary random process ξ_t satisfying the condition:

$M\{(\xi_{t+\tau} - \xi_t)^2\} \longrightarrow 0$ for $\tau \longrightarrow 0$, it is necessary and sufficient that it admits the representation:

$$F_\xi(\omega) = \iint_{-\infty}^{\infty} e^{it\omega} dF_\xi(\omega),$$

where $F_\xi(\omega)$ is an arbitrary non-negative bounded monotonically non-decreasing function that is continuous on the left.

$F_\xi(\omega)$ is called a spectral function if it is absolutely continuous and

$$F_\xi(\omega) = \iint_{-\infty}^{\infty} S(u)du,$$

where $S\ (u)$is the spectral density of the process ξ_t.

Obviously, a special case of a spectral function (for example, RGB) is the Fourier transform. The Fourier transform with a shift (the exact value does not coincide with its magmatic expectation) is commonly called the Laplace transform. Thus, the visualization by columns is the inverse Laplace transform or the conjugate view with the Laplace transform in a Banach space.

It is obvious that the tasks of visualizing the DSM can be reduced to a parameterized model of a white-noise random process [27], section 1.3 considering that $\frac{p}{N} = \mu \to 0$, where the number of blocks $p = const$, μ is a small parameter. Analysis of the dynamics of interactive visualization under random external user influences is reduced to the study of probabilistic and statistical properties of solutions of systems of differential equations perturbed by random processes (stochastic control). The system of differential equations for a parametrized model of a white-noise random process has the form:

$$\dot{x_t} = a(x_t, u(t), t) + \sum(x_t, t)\frac{1}{\mu}\xi_{\frac{1}{\mu^2}}; x_0 = x(0).$$

The problem of visualizing the DSM is a stationary linear one (see section 1.3 for more details in [27]) and therefore is not very interesting from the mathematical point of view.

The development of metaphors for visualization and interaction conjugated with the mathematical model also has a certain value. Visualization by a column is not such an elementary metaphor, taking into account the block approach. In addition, it gives rise to another metaphor – the integral metric for partial derivatives as an alternative to the vector field.

6. Discussion of development prospects for solving the problem of stochastic visual information control

In the examples under consideration, the perturbation in the parameterized model of a low-noise random process is related to the amount of data: computational accuracy, the length of the task queue, and the length of messages. From the point of view of automatic control, the task of visualizing the DSM is stationary linear task and therefore it not very interesting from the point of view of mathematics. However, the block approach itself is promising, because there are more complex tasks, for example the development of simulators with feedback, the task of detection on a height map, the consideration of which we will begin with the task of recognizing infant gestures.

6.1. Infant gesture recognition task

There is a considerable amount of work on the use of neural networks for recognizing behavioral patterns in time series, including for experiments, but their use does not guarantee that the problem is solved correctly, especially in cases where validation is subjective in nature. On the other hand, the idea of combining artificial and mathematical intelligence is tempting. In addition to the already mentioned noise reduction model, it is worth noting a new trend in solving DU, especially inverse problems: neural Fourier and Kolmogorov-Arnold operators. In fact, we will consider the Cauchy problem, that is, under what conditions the problem of recognizing baby gestures has a solution.

A multi-leaf skeleton is defined as a multi-leaf shape (a flat shape with self-intersections) [28]. A nontrivial problem will be considered when the multi-leaf skeletons of infants are anatomically similar.

The midline of a plane figure is a set of interior points of the figure, each of which has at least two nearest boundary points. Solutions to the traveling salesman problem for Delaunay graphs (Euclidean minimum spanning tree [18]) are known and Delaunay graphs (Voronoi diagrams) are also used to recognize gestures (it is assumed that the distance between the nodes of the graph (multi-sheet skeleton) does not change). An important assumption for solving the problem is that the node of the graph (joint) has an area (a problem with uncertainty). In this case, the midline s obtained as a union of straight and parabolic segments of a plane figure, for example, for the elbow joint it is schematically shown in Figure 9.

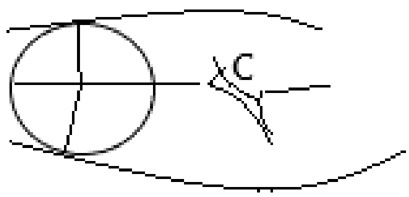

Figure 9: The median axis of the joint (c) is a concave quadrilateral, the tops of which are connected by a parabolic segments

Numerical, sufficiently accurate construction of the subdifferential in three-dimensional space, as well as the median axis, is problematic, so another task will be considered. Namely, the task of finding a list of eigenvalues of linearly independent subsets (neighborhoods of different joints) for a height map to which the Laplace transform is applied. If two distributions of a healthy and potentially sick infant are separated, then they must have a different set of transfer functions (gestures), and therefore eigenvalues. The distribution is a height map that changes over time. We take two frames that are close in time, subtract one from the other and get a matrix with a large number of" zero " elements, which has a block structure. Apply the Laplace transform to the block. We construct a transfer function from it, which exists by Sinkhorn's theorem [21]:

If A is a matrix $n \times n$ with strictly positive elements, then there are diagonal matrices D_1 and D_2 with strictly positive diagonal elements such that $D_1 A D_2$ is a doubly stochastic matrix. The matrices D_1 and D_2 are unique modulo multiplying the first matrix by a positive number and dividing the second matrix by the same number. [21].

A simple iterative method for approximating a dual stochastic matrix is to scale each row and each column of A in turn to sum to one.

Since the problem on spanning graphs with uncertainty is considered, it may not be necessary to take the difference of frames; it is sufficient to find the eigenvalues for an infinite sequence of frames (for each frame separately).

It can be concluded that the task of recognizing baby gestures is promising in terms of solution, although the known mathematical apparatus is clearly not enough.

The authors are also interested in other problem statements related to the application of stochastic semantics in the field of visualization, which are both applied and theoretical in nature. For example, the Ito formula for graphic filters (parallel filtering of data), consideration of the wave equation of rendering as a parameterized model of a white-noise random process (geometric solution).

The authors intend to work closely with the steady-state perturbed processes in the field of visualization taking into account the human factor, starting with the implementation of simple one-parameter tests (graphical filters), the number of which should be sufficiently large so that it is possible to calculate hidden dependencies (conditional probabilities) between parameters, for example, using the Kolmogorov-Arnold theorem or covariance LSM. Let's try to reduce the problem under consideration to the solution of an ODE system with interference and non-linearity of the filter composition type. (Although, in the field of visualization, it is preferable to consider Gaussian perturbed processes, that is, partial differential equations, since the Fourier transform and bursts are generally accepted in this field). In particular, it is necessary to put forward plausible hypotheses in order to determine the context, quality, cognitiveness and perceptivity of information that have a subjective connotation.

6.2. Methodology of parameter optimization in reactive computing for professional performance assessment

Interactive visualization can be considered as a random process with an important property of asymptotic convergence. Therefore, for a professional user, changing a certain parameter should tend to the optimal value.

6.2.1. Reactive computing and parallel data filtering

Data filtering is considered a special case of reactive computing. Data filtering is any operation on data that changes its quantity. Therefore, adding objects and detailing an image are filters, but suppressing noise in an image is not (white noise is not an interference). The most widespread is single-parameter data filtering the so-called slicing [9], for example, sections by a plane (sphere), or isosurfaces. Thus, parallel sections by a plane have the form:

$$\sum_{i} Mf(Ax + By + Cz \in [d_i, d_{i+1}]) = f$$

where f is a spectral function defined at the nodes of a grid (graph), not necessarily regular, i is a numerical parameter (number of blocks)similar to time, a fixed length interval and Δd determines the interferenceassociated with the accuracy of the calculation (measurement) and the choice of the number of blocks determinative calculations, for example, interaction implemented using a slider. If we go to the limit with respect to i , which tends to infinity, we get a generalization of the Ito integral through the probability limit (as noted, in the case of three-dimensional visualization, it is more convenient to consider the convergence of the fundamental sequence in a separable Banach space).

Let us consider parallel data filtering. From the point of view of parallel computing, belonging to a certain interval is a sample, which is convenient to implement as a pipeline. “Pipe” is a standard construction in the programming language being developed. Then the data must be transferred to the client computer and be displayed on the screen, this display must be continuous in a sense, which is the main topic of this section. In essence, this approach is a formalized generalization of the MVC (Model-View-Controller) architecture. In particular, the observer pattern in the context of this work is a stationary perturbed process characterized by the equiprobability of messages (reactions). In object-oriented programming, the most natural way to implement reactive computing is to add reactions to the methods and fields of objects that automatically recalculate values, and other reactions depend on changes in these values. In order to optimize performance, graph reduction (syntax tree, Petri net) is desirable; for this, it is necessary to store history, which leads to additional overhead (noise). In order to simplify the model, graph reduction will not be considered for now, in addition, data filtering is a higher-level reduction. If we compare the share of reactive computations with the share of parallel computations, and the execution of reactions with the transmission of messages, then in both cases the commonality of mathematical models is obvious, in particular the parameterized model of a white-noise random process, and therefore the commonality of syntax in the programming language. In the MVC architecture, using reactive programming, it is possible to implement automatic display of changes from Model to View and vice versa from View to Model. As noted, parallel (distributed) computations and visualization introduce their own interference.

6.2.2 Ito formula for plane section filter

In practice, a series of cross-sections with a plane parallel to one of the coordinate planes is most widely used, due to the simplicity of sampling. Of course, for the chosen values, one could apply the Laplace transform (which is infinitely deferential), but the authors intend to reduce the problem to an ODE. Consider the perturbed process by setting, for example, $\Delta d = \Delta\delta z$, so that the perturbed plane passes through the middle of the parallel plane. Since there are two possible intersections, it turns out that the order of variables is important: the resulting set will be open in x and closed in y or vice versa. For simplicity, we will consider a lattice (a regular grid) with the amount of data $f = N^3$. The purpose of this analysis is to estimate the amount of data, for example, in order to reserve memory on the GPU, when applying a cross-section plane filter (in the general case, an arbitrary filter). Obviously, two options are possible: projection бof the nearest neighboring nodes on the perturbed plane (filter), which belonged to the neighborhood Δz above the plane and in the middle of the plane. In the first case, the additional amount of data is N, which is similar to the Ito integral, and in the second case it is $2N$, which is similar to the Stratonovich integral. We will call this approach an expert approach.

Theorem: Let $f(N,\vec{x})$ be a perturbed random process (Ito) with respect to the amount of data. If the Ito process is a martingale, then for any filter $\varphi \epsilon C^2$ that is twice continuously differentiable $\varphi f(N,\vec{x})$ is a martingale.

Proof: Let us consider only the proof scheme. Let the filter φ be a generating operator (twice continuouslydifferentiable). Further, as for the one-dimensional Ito formula [4].

A consequence of the Ito formula is invariance with respect to the sum of integrals. An expert approachд in the case of the Ito process gives the following formula:

$$f = N^3 \to f'(B_N) = N^2 + N,$$

where N is *the ds* integral.

From which it is not difficult to suggest the Ito formula for a filter with a cross-section plane (the derivative of the filter)

$$f'(B_N) = (\vec{n}, \nabla \prod x_i)|_{x_i=N} + N.$$

Since the value *of ds*-integral is N, the amount of datax cannot be too large. For example, when the accuracy of calculations is comparable to the accuracy of arithmetic operations, a certain jump will occur in the constantly expanding open set (in this case, a neural network is said to have retrained), which may give some other interpretation of the generalized theorem of thermodynamics [15].

Now, instead of the plane section filter, consider, for example, a cylinder section. The authors' imagination is not so developed that they always use an expert approach. Is it possible to use the Jacobian (generating, characteristic (Hessian) operators) instead of the scalar product of the normal and gradient? It also seems obvious that the Jacobian is a martingale. The direction of the normal is important for visualization. Considering data filtering as differentiating a multidimensional random process with respect to a filter has certain prospects for visualizing multidimensional data (visualization with uncertainty, displaying the ds-integral) and evaluating computational methods from the point of view of SDE. Promising areas of stochastic visualization research: uncertainty visualization, stochastic rendering model (explicit feature extraction), considering a random process not by the amount of data, but, for example, by curvature (for example, a perturbed surface of rotation is given, it is necessary to find a perturbed axis of rotation).

6.2.3. Basic definitions and demonstration examples in the methodology of parameter optimization

In reactive computing, changing the interval length will lead to automatic recalculation of the function, for example, at the nodes of the plane. The formalization is based on the analogy that reactive computing can be considered as a particular solution of a differential equation or a phase trajectory in which the initial data (parameter values) change. (The authors have previously actively used the concept of a program trajectory [29]). Thus, parallel sections of a plane are a set of perturbed trajectories, where each plane is a set of perturbed trajectories. We emphasize once again that discrete time in this case is the number of planes (the number of blocks). The trajectory taking into account events will be denoted by *f(·,ω)*, an example of a stochastic trajectory of a program is shown in Figure 10. For comparison, Figure 11 shows a program trajectory (reachability set) that explicitly depends on time - *f(·)* [30].

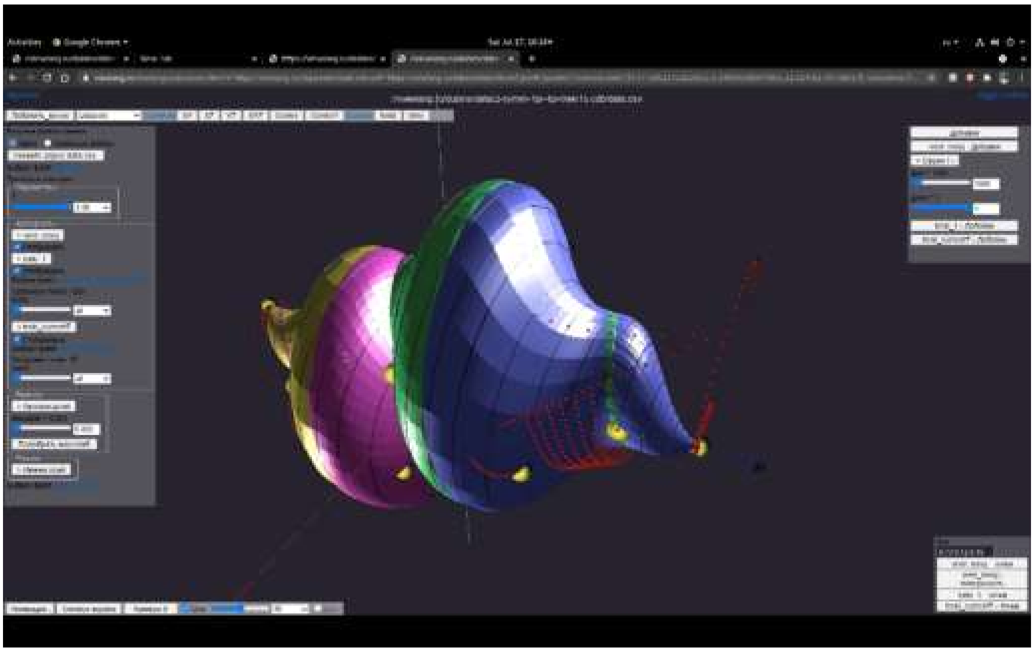

Figure 10: Stochastic trajectory of a program

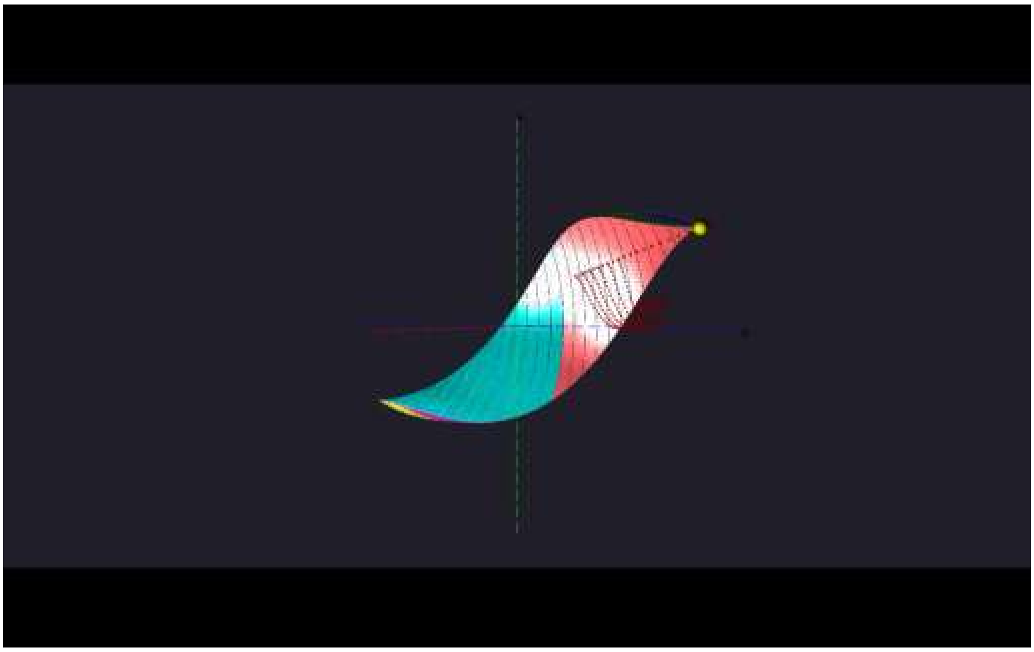

Figure 11: time-dependent trajectory of a program (reachability sets)

It's time to move on to the definition of concepts that have a subjective connotation:

1. The context (of information) is fully described by the filter pipeline, the penalty function, and interference. Therefore, a monotonous task in the sense of a professional approach from the point of view of DS is a typical parallel pipeline.
2. Information quality is a relative, non-probability (fuzzy) entropy (in fuzzy entropy, probabilities are replaced by a membership function, the value of which is determined by an expert).
3. Cognitiveness of information is the infimum of information quality.

4. Perceptiveness of information is the supremum of information quality, for which it is assumed that the PSNR (peak signal-to-noise ratio, where noise is the root mean square error) metric will be used.

Based on the visualization examples, we will give some explanations to these definitions. In such cases, we consider a Dirichlet problem with stochastic control of the form regulator with incomplete information about the state of the system-the problem of stochastic control of visual information. It is known that the exact upper bound must be determined to solve this problem, and therefore the exact lower bound пis not interesting in the modeling process.

The PSNR metric is widely used in the field of visualization, for example, in the noise reduction model, see Figure 12 [2].

Figure 12: Original image and Noisy image, PSNR=29.38dB.

For compressed images with PSNR=40-50dB, the image is considered to be of good quality. Even at a sufficiently high noise level, a person can determine that the drawing shows a woman. Therefore, the authors defined the cognitiveness of information as the exact lower limit of information quality. However, the author is interested in perturbed processesы, so a more obvious example is the mapping of a step function.

Imagine displaying a continuous function with several local extrema on the screen, where the partition parameter is the number of intervals. Although the cognitiveness of information is not particularly interesting in terms of modeling, it is likely that the exact lower bound of information quality corresponds to the minimum of noise (variance) between the points of local extremes and their projections on intervals. Since the screen consists of pixels, the exact upper bound will be defined on the Polish space (a space homeomorphic to the complete metric space with a countably dense subset). The following arguments are aimed at lowering the exact upper bound. You can add a line thickness to the image (Lifshitz condition) so that it is dense on a finite subset (all thick intervals touch each other). Such images will be called continuous from the point of view of visual perception (perceptively continuous). The compensatory function of human visual perception is well developed, so we can assume that the average noise value is less than the line thickness. In fact, this is a Bernoulli distribution. We will call such images compensatory-continuous. An example of such images is Figure 7.

Since the PSNR metric is used for two close images, it becomes necessary to fix the image selected by the expert. We will consider the parameter values corresponding to this image to be optimal, which can then be clarified based on the test results. Another hypothesis is as follows: if a person chose a parameter value less than the optimal one, then his left hemisphere (logical thinking) is more developed, if more, then the right one. The following is an example of an expert approach in the field of visualization. The hybrid view was considered for the implementation of the DSM flyby, but was not used, since the perturbed Gaussian processes coped well with this task.

The hybrid view of the display in the center (in the direct view area) has good accuracy (core), but in the periphery, the accuracy is worse (media). This task is formulated as follows: maximizing information (visualization quality) with a fixed GPU memory size. Next, we consider fundamental sequences; assuming that there is one node of the graph in one graphic primitive. Since graphic primitives occupy different volumes, this problem statement makes sense.

Modeling is based on an expert approach:

1. The quality of visualization is equal to improbability entropy.
2. An antitone distribution is constructed, see Figure 13. where V is voxels, T is 3D-textures, P is polygonal graphics, and C is a point cloud.
3. We define the quality of visualization as a function of belonging:

$$\mu(V) = 1,\ \mu(T) = 0.9,\ \mu(P) = 0.8,\ \mu(C) = 0.7.$$

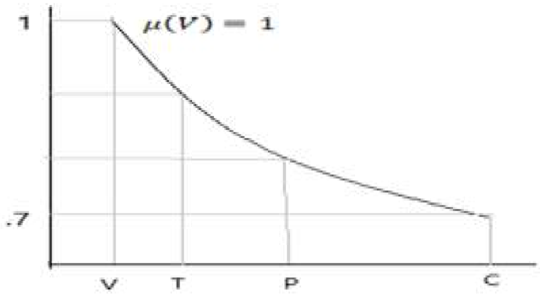

Figure 13: Antitone distributione

Consider the set of core-carrier permutations in a block model. We assume that the visualization quality of the permutation is higher if the improbability entropy (maximization principle) is correspondingly higher. The non-probability entropy is an integral characteristic of the fuzziness of a fuzzy set. It is calculated using standard formulas [31] - this is the normalized Shannon entropy:

$$\mathrm{H} = -\frac{1}{\ln \mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{p_i} \ln \mathrm{p_i},\ \ \mathrm{p_i} = \frac{\mu_{\mathrm{A}}(\mathrm{x_i})}{\sum_{\mathrm{i}=1}^{\mathrm{N}} \mu_{\mathrm{A}}(\mathrm{x_i})}$$

An intermediate goal of parameter optimization methodology is the development of single-parameter tests.

Let's consider a filter “a series of layers” Figure 14. A set of transformations of visual representation for improving the quality of perception “a series of layers” [30] has been developed and programmed. It allows one to look at the inner part of the surface, and human visual perception can approximate the surface on empty intervals (compensatory continuity is considered). It is possible to calculate a metric, which the authors called the information gap: $\sup\limits_{p} \frac{PSNR_{in}}{PSNR_{ex}}$, where p is the number of blocks (non-empty intervals), in is the inner part of the surface and ex is the outer part of the surface.

Obviously, for a single-color cylinder, three nonempty intervals are sufficient. In this way, you can define a random process (interactive visualization) that maps the number of colors and the curvature of the surface to the number of intervals and the length of the interval (which can be calculated from the number of intervals). In other words, the user is given the task to change the number of layers, which should tend to the optimal one. Thus, you can track three quasi-objective parameters: the deviation from the optimal number of layers in (you can also add plane clipping, but one parameter is better to start with), the number of iterations, which determines the convergence rate, and the binary parameter (presence) that is change of point of view.

The programming goal is to create a set of such tests. In the context of visual perception, autists know a certain number of such tests that are not sufficient for modeling: the sunrise metaphor, motion parallax, and contrast change.

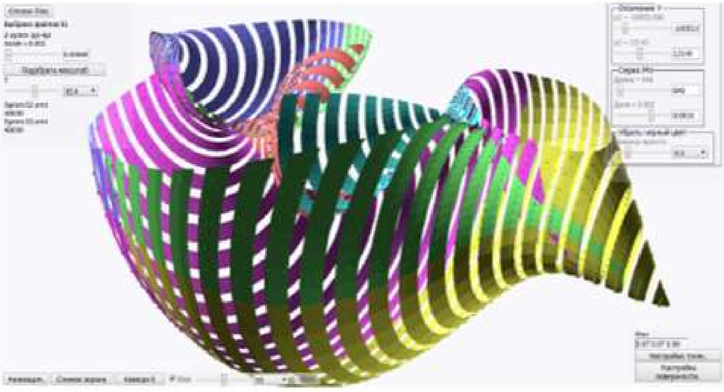

Figure 14: Visual image of the reachability set with added transformations a y-axis clipping and a series of layers along the ϕ-axis [30].

To conclude this section, we will add a couple of obvious definitions. The combinatorial function (human) in the case of independent variables (parameters) is described by the amount of information (the property of information additivity), and in the case of dependent variables by the Kolmogorov-Arnold theorem, since only continuous maps are considered. A compensatory function is a weighted sum with variable weights (a person automatically determines these weights depending on the situation, as noted above, the Bernoulli distribution can be considered).

7. Conclusion

For software verification, as well as for visualization, SDE, information theory, and signal graphs are used in this work. This approach is called stochastic semantics. It is important that the visual process is a parallel processoм (interacting sequential Hoare processes) from the point of view of programming and a random processoм from the point of view of mathematical modeling.

Considering problems related to really big data inevitably leads to the use of a block approach. In parallel computing, a block can be associated with a processor and the task of maximizing entropy (performance) can be considered. In the developed dynamic system of online visualization and parallel computing for geometric parallelization, it is possible to implement and compare a stationary random process and a steady-state random process, which have different analytical solutions. This allows us to conclude that the proposed implementation of the stationary process has a certain novelty.

Not much has been done in the field of visualization verification – grid visualization is proposed, which is considered as a parameterized model of a white-noise random process. The authors are also interested in other problem statements related to the application of stochastic semantics in the field of visualization. I would particularly like to mention the research series on generalized computational experiment [28].

Of course, this work cannot be considered complete, but the direction that the authors called stochastic semantics is obviously promising. The authors intend to deal closely with the established perturbed processes in the field of visualization, including taking into account the human factor (the outline of formalization is given in the form of a discussion for the detection problem and methods for optimizing parameters in reactive computing for evaluating professional activity).

References

1. Croitoru F-A., Hondru V., Tudor Ionescu R., and Shah M. Diffusion models in vision: a survey, // IEEE Transactions on pattern analysis and machine intelligence 14(8): 2022. P. 1-25.

2. L´ezoray O. and Grady L Image Processing and Analysis with Graphs: Theory and Practice, CRC Press, July 2012. Graph Signal Processing and Applications.
3. Cheng-Zhong Xu, Francis C., Lau M. Optimal parameters for load balancing with the diffusion method in mesh networks. // Parallel processing letters, Volume 4, 1994 P. 139-147.
4. Oksendal B. (2003). Stochastic differential equations: an introduction with applications. Berlin: Springer. ISBN 3-540-04758-1.
5. Fathi A. (2009). Weak KAM theory in lagrangian dynamics. Cambridge studies in advanced mathematics.
6. Manakov D., Averbukh V. Verification of visualization // Scientific visualization 2016. Quarter 1. Volume 8. N: 1. P. 58 - 94. [In Russian]
7. Lakoff G. The contemporary theory of metaphor // Metaphor and thought. (2nd ed.). Cambridge: Cambridge university press, 1993, P. 202-251.
8. Scott D.S. Data types as lattices // Proceedings of the international summer institute and logic colloquium, Kiel, in Lecture notes in mathematics. Springer-Verlag. 499. P. 579-651.
9. Averbukh V. L., Manakov D. V. Analysis and visualization of "big data" // Proceedings of the International Scientific Conference "Parallel Computing Technologies" (2015). Yekaterinburg, March 31 - April 2, 2015. Chelyabinsk, SUSU Publishing Center. 2015. pp.332-340. [In Russian]
10. Vasev P. A. Visualizing the operation of the parallel task planning algorithm // GrafiKon 2023: 33rd International Conference on Computer Graphics and Machine Vision, September 19-21, 2023, Moscow: trudy, pp. 341-353. [In Russian] DOI: https://doi.org/10.20948/graphicon-2023-341-353
11. Kotov V.E., Problems of parallel programming development // Proceedings of the All-Union Symposium "Prospects for System and Theoretical Programming". Novosibirsk, 1979. pp.58-72. [In Russian]
12. Biberstein O., Buchs D., and Guelfi N. //Object-oriented nets with algebraic specifications: The CO-OPN/2 formalism. / Agha G., De Cindioand F, and Rozenberg G. editors, Advances in Petri nets on object-orientation, LNCS. Springer-Verlag, 2001
13. Kudrin K. A., Kovartsev A. N., Prokhorov S. A. Methods of debugging automation in graph-symbolic programming technology //Collection of scientific papers "Information systems and technologies", Samara, 1996. pp. 75-79. [In Russian]
14. Averboukh Y. Lattice approximations of the first-order mean field type differential games. Nonlinear Differ. Equ. Appl. 28, 65 (2021). DOI: 10.1007/s00030-021-00727-2
15. Stratonovich R.L. Theory of Information, Moscow: Sov. radio, 1975. , 424 p. [In Russian]
16. Baimuratov I., Nguyen T., Golchin R., Mouromtsev D. Developing non-empirical metrics and tools for ontology visualizations evaluation and comparing. Scientific Visualization. 2020. Vol. 12. N. 4. P. 71-84. DOI: 10.26583/sv.12.4.07.
17. Shih M., Zhang Y., Ma K.-L., Sitaraman J., Mavriplis D. Out-of-Core Visualization of TimeVarying Hybrid-Grid Volume Data // IEEE Symposium on Large Data Analysis and Visualization. 2014. P. 93 – 100.
18. Neznakhina E.D. PTAS for the Min-k-SCCP problem in a Euclidean space of arbitrary fixed dimension //Proceedings of the Institute of Mathematics and Mechanics of the Ural Branch of the Russian Academy of Sciences, 2015, vol.21, no. 3, pp.268-278.
19. Datta N., Jozsa R., Dorlas T. Benatti F. Properties of Subentropy, // J. Math. Phys. 55 (2014) 062203. doi: 10.1063/1.4882935
20. Teplov A.M. On one approach to comparing scalability of parallel programs / / Computational methods and programming. 2014. Vol. 15. Issue 4. pp. 697-711
21. Rozenberg V. L. Spherical block model of dynamics and seismicity of the lithosphere: current state and development prospects // Modern methods of seismic hazard assessment and earthquake prediction: III All-Russian Scientific Conference with International Participation, October 25-26, 2023, Moscow: materials. Moscow: ITPZ RAS, 2023. pp. 224-228.

22. Sinkhorn, Richard. (1964). A relationship between arbitrary positive matrices and doubly stochastic matrices // Ann. Math. Statist. 35. P. 876–879. doi:10.1214/aoms/1177703591

23. Zenkin A. A. Cognitive computer graphics / Ed. D. A. Pospelov. Moscow: Nauka, 1991. 192 p. [In Russian]

24. Cui Q., Ward M., Rundensteiner E., and Yang J. Measuring data abstraction quality in multiresolution visualizations // IEEE TVCG, 12(5): 2006. P. 709–716,

25. Yang J., Ward M. and Rundensteiner E. Hierarchical exploration of large multivariate data sets // Data visualization: The state of the art 2003, P. 201–212.

26. Manakov D.V. Data abstraction models: sampling (parallel coordinates), filtering, clustering // Scientific visualization 2019, Q. 1. Vol. 11. N: 1. pp. 139-176, [In Russian] DOI: 10.26583 / sv.11.1.11

27. Kuznetsov D.F. Some issues of the theory of numerical solution of Ito stochastic differential equations // Differential equations and control processes. 1998. N 1. 367 p. [In Russian]

28. Mekhedov I.S. Multi-sheeted plane figure and its median axis // News of universities. Mathematics. 2011. N 12. P. 42–53. [In Russian]

29. Vasev P., Bakhterev M., Manakov D., Porshnev S., Forghani M. // On expressiveness of visualization systems' interfaces/. Scientific visualization/ 2022 14.5: P. 77 - 95, DOI: 10.26583/sv.14.5.06

30. Vasev P.A., Fedotov A.A. Visualization of three-dimensional reachable set for the Dubins car / Proceedings of the All-Russian Conference with international participation “Control Theory and Mathematical Modeling”, dedicated to the memory of Professor N.V. Azbelev and Professor E.L. Tonkov, Izhevsk. 2020, P. 264–265.

31. Kononyuk A.E. Discrete-continuous mathematics. (Sets (fuzzy)). - In 12 books. Book 2, part 2 - K.: Education of Ukraine. 2012. 452 p. [In Russian]

32. Alekseev A.K., Bondarev A.E., Pyatakova Yu.S. On the use of canonical decomposition for visualization of the results of parametric calculations // Scientific visualization, 2023, Kv.4.Vol. 15. N: 4. P. 12 - 23, DOI: 10.26583 / sv. 15. 4. 02