

DISCOVERIES ON DEMAND

BORIS STILMAN

University of Colorado Denver & STILMAN Advanced Strategies, Denver, USA.

ABSTRACT

This paper is the next logical step in discovering the algorithm for inventing new algorithms, the Algorithm of Discovery. This research is within the scope of investigation of the structure of the Primary Language of the human brain as introduced by J. von Neumann in 1957. In this paper, we revise our hypothesis that the Primary Language is a collection of algorithms crucial for survival and development of humanity including two major components, the algorithm optimizing warfighting, Linguistic Geometry (LG), and the Algorithm of Discovery. We suggest that the Primary Language is the Language of Visual Streams (“mental movies”) that empower directly both of those components as well as all the human symbolic languages and sciences. The Algorithm of Discovery is based on multiple thought experiments, which manifest themselves via visual streams. Those streams can run concurrently and exchange information between each other. The streams may initiate additional thought experiments, program them, and execute them in due course. The streams may operate employing the algorithm of “a child playing a construction set” that includes a construction model, a construction set and the Ghost. The streams are focused employing several types of reasoning including proximity and mosaic reasoning. In this paper we introduce a concise discovery program generated as a result of execution of the Algorithm of Discovery. A discovery program consists of a recorded series of thought experiments in a way that every experiment either builds new theme of visual streams or utilizes themes built previously. A series of themes of streams is the key for making discoveries.

Keywords: algorithm of discovery, artificial intelligence, linguistic geometry, mosaic reasoning, primary language, proximity reasoning, visual stream.

1 THE PRIMARY LANGUAGE

In 1957, J. von Neumann hypothesized existence of the Primary Language of the human brain [1]. He suggested that the external language (including multiplicity of natural languages as well as mathematics and computer science) that we use in communicating with each other may be quite different from the internal language used for computation by the human brain. He argued that we are still totally unaware of the nature of the Primary Language for mental calculation. He writes: “It is only proper to realize that [human] language is largely an historical accident. The basic human languages are traditionally transmitted to us in various forms, but their very multiplicity proves that there is nothing absolute and necessary about them. Just as languages like Greek or Sanskrit are historical facts and not absolute logical necessities, it is only reasonable to assume that logics and mathematics are similarly historical, accidental forms of expression. They may have essential variants, i.e., they may exist in other forms than the ones to which we are accustomed. The language here involved may well correspond to a short code in the sense described earlier, rather than to a complete code [in modern terms, he means high-level vs. low-level programming languages]: when we talk mathematics, we may be discussing a secondary language, built on the primary language truly used by the central nervous system.” Over the last 50 years since J. von Neumann hypothesized existence of the Primary Language the nature of this language has been unknown.



This paper is part of the Proceedings of the New Forest Conference on Complex Systems 2016
(Complex Systems 2016)
www.witconferences.com

© 2016 WIT Press, www.witpress.com

ISSN: 1755-7437 (paper format), ISSN: 1755-7445 (online), <http://www.witpress.com/journals>

DOI: 10.2495/DNE-V11-N4-495-507

Our original hypothesis was that the Primary Language is a collection of major algorithms crucial for survival and development of humanity, the underlying “invisible” foundation of all the modern languages and sciences [2]. We suggested that those components include Linguistic Geometry (LG) [3–16, 17–22], the algorithm of efficient warfighting, and the Algorithm of Discovery. This hypothesis was further developed in [2, 23–29]. Extensive investigation of the mental visual streams as a foundation of the Algorithm of Discovery and its application to rediscovering LG led us to observe powerful universal nature of those streams. We realized that our original hypothesis about the Primary Language should be revised. It appears that the Primary Language is the Language of Visual Streams (Section 2). In such capacity, it certainly serves as the language utilized directly by both algorithms, LG and the Algorithm of Discovery. Moreover, it is the underlying foundation of all the sciences and human languages as it should have been according to J. von Neumann [1].

Following J. von Neumann, we call it a language although it is not a language in mathematical sense, that is, it is not a set of strings of symbols. The Primary Language is the engine with the ability to generate visual streams and focus them in a desired direction. Visual streams serve as the driving force behind the Algorithm of Discovery for constructing new algorithms and, essentially, making discoveries. Visual streams empower LG for generating winning strategies for opposing games.

We suggest that visual streams drive generation for all the human symbolic languages including sound, pictorial, and written languages. This generation happens as real-time narration to running visual streams. Essentially, when talking, drawing, and writing, we describe what we “see” in visual stream using symbolic “summaries”. For this purpose a stream can play real or imaginary world scenarios in real time or faster than real time. It can also branch into other streams, morph itself, zoom in and out, etc. Though these activities are controlled to support narration, the stream (or streams) has a lot of freedom of direction. It is likely that this freedom is not just a random choice but influenced by external features such as emotions. In addition, a visual stream is the key for translation from one human language to another. Essentially, every phrase or write-up in natural language is accompanied by a respective visual stream which is initiated after being retrieved from the mental archive. Then translation of this phrase to a different language is a narration of the visual stream employing this language. The Language of Visual Streams certainly supports all the semantics which is missing in conventional translation algorithms.

With respect to the Algorithm of Discovery the streams control is substantially different from the one utilized for generation and translation of languages because its’ purpose is not simply a narration. In this case, the streams serve as construction tools of the discovery program whose execution leads to the discovery (Sections 6). It is even more interesting, following the J. von Neumann suggestion, to consider how the visual streams engine empowers all the sciences. Indeed, in our investigation of the Algorithm of Discovery, we applied it to rediscovering various components of LG, 3D structure of DNA, and the Theory of Special Relativity [2, 23–29]. In all cases, those re-discoveries happened first in the form of visual streams that eventually were transformed (recorded) into the conventional symbolic algorithms. It is reasonable to suggest that those visual streams introduced in [2, 23–29] is the “primary language” of LG, 3D structure of DNA, and the Theory of Special Relativity and, likely, all other sciences. The role of the Language of Visual Streams in empowering symbolic languages and sciences will be further investigated in our future research. Additional details are also provided in Section 3 in the description of the symbolic shell.

We will provide more details about visual streams in the following sections. The purpose of this paper is twofold, to make the next step in discovering the Algorithm on Discovery and to better understand the nature and usage of the visual streams as components of the Primary Language.

2 TOWARDS ORDINARY DISCOVERIES

What if discoveries are produced routinely as an output of computer programs? What a leap this would mean for humanity? Approaching this time of making discoveries on demand is the purpose of our efforts.

In a number of papers, we have been developing a hypothesis that there is a universal Algorithm of Discovery driving all the innovations and, certainly, the advances in all sciences [2, 23–29]. All the human discoveries from mastering fire more than a million years ago to understanding the structure of our Solar System to inventing airplane to revealing the structure of DNA to mastering nuclear power utilized this algorithm. The Algorithm of Discovery should be a major ancient item “recorded” in the Primary Language due to its key role in the development of humanity. This line of research involved investigating past discoveries and experiences of construction of various new algorithms, beginning from those, which we were personally involved in [2–36, 38–49].

Thought experiments allow us, by pure reflection, to draw conclusions about the laws of nature [30]. For example, Galileo before even starting dropping stones from the Tower in Pisa used pure imaginative reasoning to conclude that two bodies of different masses fall at the same speed. The Albert Einstein’s thought experiments that inspired his ideas of the special and general relativity are known even better [31, 32]. The efficiency and the very possibility of thought experiments show that our mind incorporates animated models of the reality, e.g., laws of physics, mathematics, human activities, etc. Scientists managed to decode some of the human mental images by visualizing their traces on the cortex [30, 33]. It was shown that when we imagine a shape “in the mind’s eye”, the activity in the visual areas of the brain sketches the contours of the imagined object; thus, mental images have the analogical nature. It appears that we simulate the laws of nature by physically reflecting the reality in our brain. The human species and even animals would have had difficulty to survive without even minimal “understanding” of the laws of environment. Over the course of evolution and during development of every organism, our nervous system learns to comprehend its environment, that is, to “literally take it into ourselves” in the form of mental images, which is a small scale reproduction of the laws of nature. Neuropsychologists discovered that “we carry within ourselves a universe of mental objects whose laws imitate those of physics and geometry” [30]. In [2], we suggested that we also carry the laws of the major human relations including the laws of optimal warfighting. The laws of nature and human relations manifest themselves in many different ways. However, the clearest manifestation is in perception and in action. For example, we can say that the sensorimotor system of the human brain “understands kinematics” when it anticipates the trajectories of objects. It is really fascinating that these same “laws continue to be applicable in the absence of any action or perception when we merely imagine a moving object or a trajectory on a map” [30]. This observation, of course, covers actions of all kinds of objects, natural and artificial. Scientists have shown that the time needed to rotate or explore these mental images follows a linear function of the angle or distance traveled as if we really traveled with a constant speed. They concluded that “mental trajectory imitates that of a physical object” [30].

Our hypothesis is that the Algorithm of Discovery is based not on formal logic but on the visual streams, that is, mental imaginary movies which run in our brain. (By the way, LG is based on those streams as well [2, 23–26].) This is how it may work. Within the brain, the visual streams run consciously and subconsciously and may switch places from time to time (in relation to conscious/subconscious use). We may run several visual streams concurrently, morph them, and even use logic for such morphing, although this use is auxiliary. Then, we mentally tag some of the objects, actions, and events shown in the movie and create the so-called symbolic shell around the main visual stream.

This shell eventually becomes a standard symbolic algorithm that can be communicated to others employing familiar language, logic, mathematics, etc.

We named this approach “visual reasoning” [26]. According to our original hypothesis of visual streams we assumed that the “visual” component (including pattern recognition) is, in general, pretty sophisticated, while the “reasoning” component is relatively simple. We also assumed that the full scale mental visibility (with pattern recognition) is rarely used in discoveries, while the limited visibility can be simulated with a reasonable effort. The “reasoning” component is certainly within the scope of the modern software development. Our assumption of the need for “standard” visibility was based on the following. When we introduced mental visual streams we assumed that we “mentally” observe events happening within those streams, that is, we see what is going on. And, for all of us observing or seeing something means recognizing it. Thus, as part of our original hypothesis, we assumed that pattern recognition should be an inherent part of those visual streams. In other words, we assumed that in order to simulate and animate the natural or artificial reality in the form of visual streams we have to understand it, and, certainly, recognize it in detail as a necessary component of this understanding. We also assumed that the same understanding and recognition should happen dynamically within the visual streams. In our revised hypothesis, we state that utilization of visual streams does not require pattern recognition at all. We utilize our brain’s innate ability to simulate outside world, that is, to “photocopy” the nature or invent and mentally construct artificial reality (which is, essentially, a modified nature) [34]. This ability goes further because we can animate simulation in the form of visual streams. In case of simulating (observing) the nature we periodically verify simulation, simply, “pixel by pixel”, employing information coming through our sensors, images, sounds, etc., by-passing conscious understanding of this information. In case of artificial reality verification is not needed. In all cases simulation of reality within visual streams happens by construction, i.e., via internal subconscious understanding of the simulated reality and, consequently, without pattern recognition. Specifically, due to the fact that this simulation is driven by the internal engine (visual stream) the “mechanics” of this engine is used to understand the simulated reality. Roughly, we do not have to look at this simulation from outside, even mentally, (and employ pattern recognition). The visual stream can analyze it from inside because the stream *is* the one who created it. For example, to realize similarity of two different visual streams we should recall that those streams are driven by the specific algorithms each, and these algorithms are available to the streams engine. Thus, to realize similarity of the streams we have to realize similarity between the algorithms driving them which in its turn can be done by their comparison (or mutual analysis). We conclude that visual reasoning is essentially a mental construction based reasoning which does not involve pattern recognition.

Our approach to discovering the Algorithm of Discovery is analogous to an attempt to understand the algorithm of a program while watching its execution. In addition to this Algorithm, we are trying to discover the instruction set of the “computer” running this program, i.e., the means of the human brain to run it. In our investigation we assume that this program simulates sequences of events while its interface includes color movies demonstrating visually those events. With multiple published introspections of great scientists we can recreate clips from various movies, that is, their imaginary thought experiments. What really helps is the assumption that all those movies were “demonstrated” by the programs running essentially the same algorithm, the Algorithm of Discovery. With our own developments in LG, we have additional power of asking what-if questions via morphing our own movies and getting answers by watching those morphed movies until the very end. Unfortunately, we do not have this power with the discoveries of other scientists. However, we could utilize their published and oral introspections.

3 THE ALGORITHM OF DISCOVERY

In this section, we briefly summarize the results introduced in [2, 23–29]. The Algorithm of Discovery operates as a series of thought experiments, which are performed via imaginary animated movies (plays), named visual streams. These streams may or may not reflect the reality. In addition, those streams serve as the only interface of the Algorithm of Discovery. This interface is constructive, that is, visual streams could be morphed in the desired direction.

The input to the Algorithm is also a visual stream, which includes several visual instances of the object whose structure has to be understood or whose algorithm of construction has to be developed. Sometimes, the object is dynamic, that is, its structure is changing in time. Then the input visual stream includes this visual dynamics. As a rule, neither the structure of the object nor the details of the dynamics are present in the stream. It simply replicates (mimics) the natural or imaginary phenomenon. The task of the Algorithm of Discovery is to understand its structure including dynamics and/or develop an algorithm for reconstructing this object including its changes in time. This understanding happens in several stages. Importantly, it always ends up with the process of actual reconstruction of the object employing the construction set developed by the Algorithm on the previous stages. If the Algorithm investigates a natural real life object this imaginary reconstruction may be totally unrelated to the construction (replication) utilized by the nature. Usually, this reconstruction process is artificially developed by the Algorithm of Discovery with the only purpose to reveal the structure of the object. However, if the algorithm of natural replication is the goal of discovery, then the Algorithm of Discovery will employ a set of different visual streams (or different themes) to reveal the relevant components utilized by the nature [28, 35, 36].

All the visual streams are divided into classes, Observation, Construction, and Validation. They usually follow each other but may be nested hierarchically, with several levels of depth.

The visual streams operate in a very simple fashion similar to a child construction set. The Construction stream utilizes a construction set and a mental visual prototype, a model to be referenced during construction. This is similar to a list of models pictured in a manual (or a visual guide) enclosed to every commercial construction set. It appears that all the thought experiments in LG related to construction investigated so far, utilized those manuals. Imagine a child playing a construction set. He needs a manual to construct an object by looking constantly at its picture included in this manual. This model comes from the Observation stream as its output. It is not necessarily a real world model. It is not even a model from the problem statement. It is created by the Observation stream out of various multiple instances of the real world objects by abstraction, specifically, by “erasing the particulars”. A final version of the object constructed by the Construction stream should be validated by the Validation stream.

The Algorithm of Discovery initiates the Observation stream, which must carefully examine the object. It has to morph the input visual stream and run it several times to observe (mentally) various instances of the object from several directions. Often, for understanding the object, it has to observe the whole class of objects considered analogous. If the object is dynamic (a process) it has to be observed in action. For this purpose, the Observation stream runs the process under different conditions to observe it in different situations. The purpose of all those observations is erasing the particulars to reveal the general relations behind them. A good example of multiple observations of processes is related to the thought experiments with various objects with respect to the inertial reference frames when discovering the Theory of Special Relativity. This includes experiments with uniformly moving ships, trains, experiments with ether as well as experiments for catching a beam of light [29, 31, 32]. Once the relations have been revealed, a construction set and a visual model have to be constructed by the Observation stream. Both are still visual, i.e., specific, – not abstract. However, they should visually represent an abstract concept, usually, a class of objects or processes,

whose structure is being investigated. For construction, the Observation stream utilizes the Construction stream with auxiliary purpose (which differs from its prime purpose – see below). Note that the model construction is different from the subsequent reconstruction of the object intended to reveal its structure. This model may differ substantially from the real object or class of objects that are investigated. Its purpose is to serve as a manual to be used for references during reconstruction. Various discoveries may involve a series of models.

When the model and the construction set are ready, the Algorithm of Discovery initiates the Construction stream with its prime purpose. This purpose is to construct the object (or stage the process) by selecting appropriate construction parts of the set and putting them together. If an object has a sequential nature the construction also takes place sequentially, by repetition of similar steps. If multiple models have been produced the final object construction can also be considered as a model construction. At some point of construction, the parts are tagged symbolically and, in the end, visual reasoning with symbolic tagging turns into a conventional symbolic algorithm to be verified by the subsequent Validation stream.

Models and construction sets may vary significantly for different problems. Construction of the model begins from creation of the construction set and the relations between its components. Both items should be visually convenient for construction. The Algorithm of Discovery may utilize a different model for the same object if the purpose of development is different. Such a different model is produced by a different visual stream.

In many cases the Algorithm of Discovery employs “a slave” to visually perform simple tasks for all types of visual streams. This slave may be employed by the Construction stream to move around, to observe construction parts, and put them together. More precisely, imagine a child playing a simplistic construction set. To avoid offending children, I had named this personality a Ghost. This Ghost has very limited skills, knowledge and, even, limited visibility. The Observation stream may utilize the Ghost to familiarize itself with the optional construction set, to investigate its properties. Next, the Construction stream may use the Ghost to perform the actual construction employing those properties. Eventually, the Validation stream may use the Ghost to verify visually, if properties of the constructed object match those revealed by the Observation stream. In all cases, the Ghost is guided by the Algorithm of Discovery or, more precisely, by the respective visual streams.

As was already discussed, the initial visual model is usually guided by a very specific prototype, where the Observation stream has actually erased the particulars. However, this specificity does not reduce generality in any way. This sounds like a paradox. Essentially, every component of this model carries an abstract class of components behind it. This way visual reasoning about the model drives reasoning about abstract classes, which is turned eventually into the standard formal reasoning. This happens as follows. A visual model drives construction of the formal symbolic model so that the key items in a visual model have tags representing the respective formal model. At first, the formal model is incomplete. At some stage, a running visual stream is accompanied by a comprehensive formal symbolic shell. Running a shell means doing formal derivation, proof, etc. synchronized with a respective visual stream. While the shell and the stream are not completely synchronized, the visual stream drives execution of the shell, not the other way around. For example, a formal proof is driven by animated events within the respective visual stream. The visual streams, usually, run the creation of the visual model, the construction set and the final construction of the object several times. During those runs as a result of persistent tagging the symbolic shell appears. Multiple runs utilize the same visual components but during initial runs the synchronization of the stream and the shell is not tight. Further on, synchronization is tightened by morphing the visual model and/or adjusting symbolic derivation if they initially mismatch. Eventually, the stream and the shell switch their roles. In the end, it appears that the stream becomes the animated set of illustrations, a movie, driven by the run-

ning symbolic shell. For example, during the final runs (and only then), the visual streams, presented in [2, 23–28], are driven by the constraints of the abstract board game, the abstract set theory and/or the productions of the controlled grammars. At this point, the visual stream and the symbolic shell can be completely separated, and the visual stream can be dropped and even forgotten.

While the symbolic shell mechanism has been introduced several years ago and demonstrated in several papers, only now we realized that it provides the link between the conventional symbolic science and languages and the same knowledge “recorded” directly in the Primary Language, i.e., the Language of Visual Streams (Section 1). Existence of “other forms [of knowledge] than the ones to which we are accustomed” was suggested by J. von Neumann, [1] and Section 1. Persistent tagging leading to creation of the symbolic shell links visual streams to the conventional science and languages. However, as it stands now, this is a one-directional link. A reverse link, i.e., initiation of the visual stream by the natural language or scientific statement happens by invoking this stream from the archive (or the data base) with subsequent morphing for a better match. Note that switching from the visual stream to symbolic shell creates symbolic knowledge (even formal knowledge in case of science). However, this switch comes at a cost of depriving this knowledge from the “creative force”. Interestingly, that many years ago we realized that development of the set of sophisticated mathematical tools of LG, the Hierarchy of Formal Languages [4–15], did not increase the inventive power of this approach. Unfortunately, at that time it looked like a particular observation.

There are several means to focus visual streams in the desired direction. They include Proximity and Mosaic reasoning.

4 PROXIMITY REASONING

Proximity reasoning as a type of visual reasoning is utilized due to the need for approaching optimum for many discoveries. It is likely that all the technological inventions and discoveries of the laws of nature include “optimal construction” or, at least, have optimization components [36]. Thus, various construction steps performed by the Algorithm of Discovery require optimization, which, certainly, makes construction more difficult. As the appearance of this Algorithm is lost in millennia, for its main purpose, it could not certainly utilize any differential calculus even for the problems, where it would be most convenient. For the same reason, it could not utilize any approximations based on the notion of a limit of function. Those components of differential calculus could certainly serve as auxiliary tools. In that sense, in order to reveal the main optimization components of this Algorithm, the most interesting problems to be investigated should lack continuity compelling the Algorithm of Discovery to employ explicitly those components. Based on several case studies [27], we suggested that this optimization is performed by the visual stream that operates imaginary movement approaching a location (or area) in the appropriate imaginary space. Having such space and means, the Algorithm employs an agent to catch sight of this location, pave the way, and approach it. Contrary to the function-based approach, which is static by its nature, the Algorithm operates with dynamic processes, the visual streams. Some of those streams approach optimum (in a small number of steps); other streams show dynamically wrong directions that do not lead to the optimum and prevent the Algorithm from pursuing those directions. Both types of streams represent proximity reasoning. We suggested that proximity reasoning plays a special role for the Algorithm of Discovery as the main means for optimization.

Proximity reasoning is a type of visual reasoning. This implies that the Algorithm should reason about the space where distances are “analogous” to the 3D Euclidian distances. Roughly, when we approach something, the distance must be visually reduced, and this should happen gradually. The space for proximity reasoning should provide means to evaluate visually if the animated images representing various abstract objects approach each other or specific locations [27]. Construction of those spaces is the key component of the Algorithm of Discovery. There could be situations when an

object approaching the desired location (of an expected optimal state) hits an obstacle, i.e., another object, but the time constraints do not permit to remove this obstacle. This way, the desired location and optimal state appear to be unreachable [10, 11]. In such cases, the Algorithm may consider a total retreat to one of the initial states where time constraints are conducive for removal of the obstacle. After removal, the Algorithm may attempt to return objects to the originally desired locations which may lead to reaching the optimum.

5 MOSAIC REASONING

The name of mosaic reasoning was introduced due to the analogy of the Construction stream operation with assembling a mosaic picture of small colorful tiles [28]. Another, maybe, even more transparent analogy is known as a jigsaw puzzle when a picture is drawn on a sheet of paper and then this paper is cut into small pieces, mixed up, to be assembled later into the original picture. As Sir G. Thompson [37] pointed "... the progress of science is a little like making a jigsaw puzzle. One makes collections of pieces which certainly fit together, though at first it is not clear where each group should come in the picture as a whole, and if at first one makes a mistake in placing it, this can be corrected later without dismantling the whole group". Both analogies, the pictorial mosaic and the jigsaw puzzle, represent well the key feature of the Algorithm of Discovery construction set. However, we prefer the former because the jigsaw puzzle looks more like an assignment in reassembling a construct, a picture, which has already been created and, certainly, well known. In that sense, a tile mosaic is created from scratch, including choosing or even creating necessary tiles. In addition, a jigsaw puzzle is reassembled out of pieces based on random cuts. On the contrary, in pictorial mosaic, in many cases, every tile should have unique properties; it should be shaped and colored to match its neighbors precisely. A similar specificity is related to a group of adjacent tiles, the aggregate.

Returning to the Algorithm of Discovery, for many discoveries, the components of the construction set should be developed with absolute precision, in the way that every part should be placed to its unique position matching its neighbors. We will use the same name, the tiles, for those construction parts. If precision is violated the final mosaic will be ruined and the discovery will not happen. Though a group of tiles, an aggregate, may be configured properly, its correct placement in the mosaic may be unclear and requires further investigation. Moreover, a tile itself may have complex structure which may require tailoring after placement in the mosaic. In some cases, a tile is a network of rigid nodes with soft, stretchable links.

Mosaic reasoning may lead to the observation, construction, and validation steps of the Algorithm of Discovery operating with tiles and aggregates of tiles. Overall, mosaic reasoning requires tedious analysis of the proper tiles and their matching rules. Investigation of the matching rules is the essential task of the Observation stream. Multiplicity of those rules and their specificity with respect to the classes of construction tiles make the actual construction very complex. Selecting a wrong tile, wrong tailoring, choosing a wrong place, or incompatible neighbors may ruin the whole mosaic. The matching rules are the necessary constraints that control the right placement of the tiles. Missing one of them, usually, leads to the wrong outcome because the Algorithm of Discovery is pointed in the wrong direction.

Some of the matching rules impact mosaic locally while other rules provide global constraints. The global matching rules include the requirement of the top-down analysis and construction, the global complementarity rule, certain statistical rules, the transformation rules, etc. For many if not all natural objects and processes, their structure is not reducible to a combination of the components. Large groups of tiles, i.e., large aggregates, may obey the rules which are hardly reducible to the rules guiding placement of singular tiles. This matching rule must be understood globally first,

implemented in the mosaic skeleton construction, and, only then, reduced to the placement of the specific tiles. The rule of the global complementarity means that placement of one aggregate may determine precisely the adjacent aggregate. The global statistical rules related to the whole mosaic may reflect the relationship between major structural components, the large aggregates. If understood and taken into account by the Observation stream, they may focus the Construction stream and lead to a quick discovery. Yet another class of global matching rules is called transformation rules. This is an algorithm for reconstructing an aggregate out of another aggregate and placing this aggregate in the proper location. Applied sequentially, such a rule permits to turn an aggregate, the so-called generator, into the set of adjacent aggregates. This way the whole mosaic could be constructed. Interestingly, this type of construction may be utilized by the Algorithm of Discovery as a convenient procedure to reveal the structure of an object, while the nature may use a totally different algorithm for generating the same object.

The local matching rules include the local complementarity rule, the interchangeability rule, etc. The local complementarity means, roughly, that a protrusion on one tile corresponds to the cavity on the complementary adjacent tile. The local complementarity often manifests itself in the requirement of various kinds of symmetry within the pairs of matching construction tiles. The whole class of the local matching rules is based on interchangeability. In simple terms, if two aggregates that include several tiles are not identical but interchangeable, their internal structure may be unimportant. There are several levels of interchangeability. Two aggregates could be essentially the same, i.e., their skeletons coincide. Importantly, those skeletons must include nodes which serve as the attaching points of the aggregates to the rest of the mosaic. The notion of an internal skeleton depends on the problem domain and is specific for different types of mosaic. Another lower level of interchangeability of the aggregates does not require their skeletons to coincide. The only requirement is that the attaching points of those aggregates are identical. In all cases interchangeability means that the stream can take one aggregate off the mosaic and replace it with another. This will certainly change the picture but the whole structure will stand. We named those aggregates plug-ins.

Besides mosaic structural components that include tiles, aggregates, global and local matching rules, there is an unstructured component that we named a mosaic environment. Such environment may impact the structure of tiles, aggregates, application of matching rules, and the whole mosaic while being relatively unstructured itself.

A transparent example of application of mosaic reasoning to the discovery of the 3D structure of the DNA molecule is considered in [28].

6 PROGRAMMING DISCOVERIES

The process of discovery is a sequence of thought experiments. This sequence could be considered as “program of discovery” as follows. This program may include subprograms. The actual creators of those programs and subprograms are visual streams. A stream may schedule other streams by creating a program with experiment “calls”. It may schedule a thought experiment to be executed immediately. After completion of this experiment based on its outcome, the stream can schedule and invoke another thought experiment, etc. We suggest that such sequence of executed experiments recorded visually (as sequence of movies) is a program of discovery. However, this is not a program in conventional sense.

The program of discovery may be executed several times, each time with different outcome. This might happen due to the different input received by each of the experiments included in the program, beginning from the first one. Moreover, this program may change during execution. Indeed, the output of an experiment usually includes request for the next experiment. Different output may include request for the experiment that differs from the one recorded previously.

Besides experiments created, executed, and recorded sequentially, one after another, a stream can create a sequence of thought experiments to be executed in the future. These experiments could, in their turn, initiate new visual streams (with new experiments). The purpose, the nature, and the general outcome of those future experiments should be known to the stream created this sequence. This sequence is different from the list of procedure calls in conventional procedural (or imperative) programming. The algorithms of those “procedures”, i.e., the algorithms to be utilized by the respective thought experiments are generally unknown. The experiments are not programmed – they are staged. The actual algorithm should be developed as a result of execution of such experiment. In a sense, this is analogous to the notion of declarative programming when a function is invoked by a problem statement while the function’s body does not include an algorithm for solving this problem. The ability of a visual stream to schedule a sequence of thought experiments (with delayed execution) permits to create a nested top-down structure of visual streams with several levels of depth. Though, we suspect that the actual depth of the nested programmed experiments never exceeds two or three.

A discovery, i.e., a development of the final algorithm for the object construction is based usually on constructing a series of themes of streams. Each of those themes may, in its turn, be based on multiple experiments and may result from multiple uses of the Observation, Construction, and Validation streams. Interestingly, those themes may represent the same object, though, be totally different. The purpose of these themes is to look at the object from different prospective to reveal different properties. The themes do not appear at once. Experiments with one theme may lead to demonstrating the need for the next one. After being invoked several themes could be utilized simultaneously. The themes construction is based on the wide use of the principle of erasing the particulars. For each theme, some of the particulars of an object under investigation are erased while other particulars are emphasized.

All the thought experiments of a discovery program could be broken into several groups. It appears that a group of experiments that operate with the same theme are related to each other while they differ from those groups that operate with another theme. Once the theme has been invoked it can usually schedule experiments within itself to be executed in the future. As a rule, after being invoked the theme is used in several experiments. Then another theme could be invoked. At that time, previous theme is turned into the idle mode, which means that the stream freezes to pass control to the stream of a new theme. The experiments with different themes cannot be scheduled until those themes have actually been invoked. The problem is that until that moment the stream is not aware of their existence and even the need for them, unless this theme has already been invoked, used, and currently preserved in an idle mode. Consequently, scheduling of experiments with delayed execution is possible with the active and idle visual themes, only.

The programming system of the Algorithm of Discovery reminds remotely a conventional compiler. The output of this compiler is analogous to the executable code, i.e., the program of discovery. Different inputs of this compiler (different “source code”) produce different outputs, i.e., different programs of discovery. Almost all of them, except for one or two, are incorrect, and do not lead to the discovery. These incorrect inputs reflect our current knowledge, e.g., incorrect experimental data, incorrect theoretical principles, etc. The failure of making a discovery, verified by the Validation stream, usually causes the Algorithm of Discovery to rerun several parts of the discovery program with different inputs. The program does not have to be rerun from the very beginning. Different parts of the program may be restarted with new inputs. As we already mentioned, different inputs to the program of discovery may change this program completely. These partial reruns with new inputs lead to the effect that, eventually, the discovery program looks like a small tree with branches at different levels. Usually, only one branch leads to discovery. This tree is a recording of the history of making a discovery including failures. With correct input the program of discovery will produce and

reproduce the right branch of the tree and, this way, will produce and reproduce the discovery as many times as it will be executed.

It appears that the most sophisticated steps of execution of the Algorithm of Discovery, i.e., generation and execution of the discovery program, are those related to invoking new themes of visual streams. While reasoning within the active and idle themes requires just logic, activating a new theme requires constructing it from scratch or retrieving it from the archive of the past themes and morphing it to match current needs.

Components of the Algorithm of Discovery described in this paper (as well as those presented in [2, 23–29]) will be hypothetical until they will be verified by software implementations. A number of implementations have been initiated at the University of Colorado Denver. More advanced implementations and experiments are planned at STILMAN Advanced Strategies.

REFERENCES

- [1] Von Neumann, J., *The Computer and the Brain*, Yale University Press, 1958.
- [2] Stilman, B., Thought experiments in linguistic geometry. *Proc. of the 3d Int. Conf on Advanced Cognitive Technologies and Applications – COGNITIVE'2011*, Rome, Italy, pp. 77–83, 2011.
- [3] Stilman, B., *Formation of the Set of Trajectory Bundles, Appendix 1 to the book: On the Cybernetic Goal of Games*, by Botvinnik, M.M., Soviet Radio, Moscow (in Russian), pp. 70–77, 1975.
- [4] Stilman, B., Ierarhia formalnikh grammatik dla reshenia prebornikh zadach (Hierarchy of Formal Grammars for Solving Search Problems), Tech. Report, VNIIE, Moscow (in Russian), p. 105, 1976.
- [5] Stilman, B., A formal language for hierarchical systems control. *International Journal Languages of Design*, **1**(4), pp. 333–356, 1993.
- [6] Stilman, B., A linguistic approach to geometric reasoning. *International Journal of Computers & Mathematics with Applications*, **26**(7), pp. 29–58, 1993.
[http://dx.doi.org/10.1016/0898-1221\(93\)90049-2](http://dx.doi.org/10.1016/0898-1221(93)90049-2)
- [7] Stilman, B., Network languages for complex systems. *International Journal of Computers & Mathematics with Application*, **26**(8), pp. 51–80, 1993.
[http://dx.doi.org/10.1016/0898-1221\(93\)90331-O](http://dx.doi.org/10.1016/0898-1221(93)90331-O)
- [8] Stilman, B., Linguistic geometry for control systems design. *International Journal of Computers and their Applications*, **1**(2), pp. 89–110, 1994.
- [9] Stilman, B., Translations of network languages. *International Journal of Computers & Mathematics with Application*, **27**(2), pp. 65–98, 1994.
[http://dx.doi.org/10.1016/0898-1221\(94\)90037-X](http://dx.doi.org/10.1016/0898-1221(94)90037-X)
- [10] Stilman, B., Deep search in linguistic geometry, symp. on linguistic geometry and semantic control. *Proc. of the First World Congress on Intelligent Manufacturing: Processes and Systems*, Mayaguez, Puerto Rico, pp. 868–879, 1995.
- [11] Stilman, B., A linguistic geometry for 3d strategic planning. *Proc. of the 1995 Goddard Conference on Space Applications of Artificial Intelligence and Emerging Information Technologies*, NASA Goddard Space Flight Center, Greenbelt, MD, USA, pp. 279–295, 1995.
- [12] Stilman, B., Linguistic geometry tools generate optimal solutions. *Proc. of the 4th Int. Conference on Conceptual Structures - ICCS'96*, Sydney, Australia, pp. 75–99, 1996.
http://dx.doi.org/10.1007/3-540-61534-2_5
- [13] Stilman, B., Managing search complexity in linguistic geometry. *IEEE Transaction on Systems*, **27**(6), pp. 978–998, 1997.
<http://dx.doi.org/10.1109/3477.650058>

- [14] Stilman, B., Network languages for concurrent multi-agent systems. *International Journal of Computers & Mathematics with Application*, **34**(1), pp. 103–136, 1997.
[http://dx.doi.org/10.1016/S0898-1221\(97\)00102-8](http://dx.doi.org/10.1016/S0898-1221(97)00102-8)
- [15] Stilman, B., *Linguistic Geometry: From Search to Construction*, Kluwer Academic Publishers (now Springer), p. 416, 2000.
- [16] Stilman, B., Linguistic geometry and evolution of intelligence. *ISAST Transaction on Computers and Intelligent Systems*, **3**(2), pp. 23–37, 2011.
- [17] Stilman, B., Yakhnis, V. & Umanskiy, O., Winning strategies for robotic wars: defense applications of linguistic geometry. *Artificial Life and Robotics*, **4**(3), 2000.
<http://dx.doi.org/10.1007/BF02481336>
- [18] Stilman, B., Yakhnis, V. & Umanskiy, O., Knowledge acquisition and strategy generation with lg wargaming tools. *International Journal of Computer Intelligence and Applications*, **2**(4), pp. 385–409, 2002.
<http://dx.doi.org/10.1142/S1469026802000713>
- [19] Stilman, B., Yakhnis, V. & Umanskiy, O., Chapter 3.3. Strategies in large scale problems. In *Linguistic Geometry: From Search to Construction*, pp. 251–285, 2007.
- [20] Stilman, B., Yakhnis, V. & Umanskiy, O., Linguistic geometry: the age of maturity. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, **14**(6), pp. 684–699, 2010.
- [21] Stilman, B., Yakhnis, V. & Umanskiy, O., Revisiting history with linguistic geometry. *ISAST Transaction on Computers and Intelligent Systems*, **2**(2), pp. 22–38, 2010.
- [22] Stilman, B., Yakhnis, V. & Umanskiy, O., The primary language of ancient battles. *International Journal of Machine Learning and Cybernetics*, **2**(3), pp. 157–176, 2011.
<http://dx.doi.org/10.1007/s13042-011-0029-9>
- [23] Stilman, B., Discovering the discovery of linguistic geometry. *International Journal of Machine Learning and Cybernetics*, **4**(6), pp. 575–594, 2012.
<http://dx.doi.org/10.1007/s13042-012-0114-8>
- [24] Stilman, B., Discovering the discovery of the no-search approach. *International Journal of Machine Learning and Cybernetics*, p.27, 2012.
<http://dx.doi.org/10.1007/s13042-012-0127-3>
- [25] Stilman, B., Discovering the discovery of the hierarchy of formal languages. *International Journal of Machine Learning and Cybernetics Springer*, p. 25, 2012.
<http://dx.doi.org/10.1007/s13042-012-0146-0>
- [26] Stilman, B., Visual reasoning for discoveries. *International Journal of Machine Learning and Cybernetics, Springer*, p. 23, 2013.
<http://dx.doi.org/10.1007/s13042-013-0189-x>
- [27] Stilman, B., Proximity reasoning for discoveries. *International Journal of Machine Learning and Cybernetics, Springer*, p. 31, 2014.
- [28] <http://dx.doi.org/10.1007/s13042-014-0249-x>
- [29] Stilman, B., Mosaic reasoning for discoveries. *Journal of Artificial Intelligence and Soft Computing Research*, **3**(3), pp. 147–173, 2013.
<http://dx.doi.org/10.2478/jaiscr-2014-0011>
- [30] Stilman, B., The primary language of battles and discoveries. *International Journal of Applied Mathematics and Fundamental Informatics*, **1**, pp. 160–166, 2014.
- [31] Deheaene, S., A few steps toward a science of mental life. *Mind, Brain and Education*, **1**(1), pp. 28–47, 2007.
<http://dx.doi.org/10.1111/j.1751-228X.2007.00003.x>

- [32] Einstein, A., *On the Electrodynamics of Moving Bodies*, *Annalen der Physik*, 17, 891, 1905 (in German).
<http://dx.doi.org/10.1002/andp.19053221004>
- [33] Miller, A.L., *Insights of Genius: Imagery and Creativity in Science and Art*, Copernicus, an imprint of Springer-Verlag, 1996.
<http://dx.doi.org/10.1007/978-1-4612-2388-7>
- [34] Kosslyn, A., Thompson, W., Kim, I. & Alpert, N., Representations of mental images in primary visual cortex. *Nature*, **378**, pp. 496–498, 1995.
<http://dx.doi.org/10.1038/378496a0>
- [35] Iacoboni, M., *Mirroring People, The New Science of How We Connect with Others*, Farrar, Straus and Giroux, New York, 2008.
- [36] Watson, J.D., *The Double Helix: A Personal Account of the Discovery of the Structure of DNA*, Atheneum, New York, 1968. [Scribner Classics Edition, New York, 1996.]
- [37] Watson, J.D. & Crick, F.H.C., A structure for deoxyribose nucleic acid. *Nature*, **171**, pp. 737–738, 1953.
<http://dx.doi.org/10.1038/171737a0>
- [38] Thomson, G., *The Inspiration of Science*, Oxford University Press, London, 1961.
- [39] Astbury, W.T., X-Ray studies of nucleic acids. *Symposia of the Society of Experimental Biology*, **1**, pp. 66–76, 1947.
- [40] Botvinnik, M., *Chess, Computers, and Long-Range Planning*, Springer-Verlag, New York, 1970.
<http://dx.doi.org/10.1007/978-1-4684-6245-6>
- [41] Botvinnik, M., Blok-skema algoritma igry v shahmaty (in Russian: A Flow-Chart of the Algorithm for Playing Chess), Sovetskoe Radio, 1972.
- [42] Botvinnik, M., *Computers in Chess: Solving Inexact Search Problems*, Springer-Verlag, 1984.
<http://dx.doi.org/10.1007/978-1-4612-5204-7>
- [43] Chargaff, E., Structure and function of nucleic acids as cell constituents. *Federation Proceedings*, **10**, pp. 654–659, 1951.
- [44] Crick, F.H.C. & Watson, J.D., The complementary structure of deoxyribonucleic acid Proceedings of the Royal Society A, **223**, pp. 80–96, 1954.
<http://dx.doi.org/10.1098/rspa.1954.0101>
- [45] Foppl, O., *Introduction to Maxwell's Theory of Electricity*, B.G. Teubner: Leipzig, 1894 (in German).
- [46] Galilei, G., Dialogue Concerning the Two Chief World Systems, 1632 (tranl. by Stillman Drake).
- [47] Furberg, S., On the structure of nucleic acids. *Acta Chemica Scandinavica*, **6**, pp. 634–640, 1952.
<http://dx.doi.org/10.3891/acta.chem.scand.06-0634>
- [48] Kott, A. & McEneaney, W. (eds), *Adversarial Reasoning: Computational Approaches to Reading the Opponent's Mind*, Chapman & Hall/CRC, 2007.
- [49] Pauling, L. & Corey, R.B., A proposed structure for the nucleic acids. *Nature*, **171**, pp. 346–359, 1953.
<http://dx.doi.org/10.1038/171346a0>
- [50] Watson, J.D. & Crick, F.H.C., The structure of DNA. *Cold Spring Harbor Symposia on Quantitative Biology*, **18**, pp. 123–131, 1953.
<http://dx.doi.org/10.1101/SQB.1953.018.01.020>