Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 46 (2015) 784 – 793

Procedia Computer Science

International Conference on Information and Communication Technologies (ICICT 2014)

Revisiting Major Discoveries in Linguistic Geometry with Mosaic Reasoning

Boris Stilman[a,b,*], May Aldossary[a]

[a]*Depatment of Computer Science and Engineering, University of Colorado Denver, Denver,CO 80202, USA*
[b]*STILMAN Advanced Strategies, Denver, CO 80210, USA*

Abstract

In discovering the nature of the Primary Language of the human brain introduced by J. von Neumann, two components have been investigated. The first component is Linguistic Geometry (LG), the algorithm of optimizing war-fighting strategies. LG's universal applicability in various domains and its power in generating human-like strategies suggested that LG should be a component of the Primary Language. The second component is the algorithm of inventing new algorithms. This paper researches the role of mosaic reasoning, a component of the Algorithm of Discovery, in obtaining the key results in LG such as the algorithms for generating trajectories and zones.

Peer-review under responsibility of organizing committee of the International Conference on Information and Communication Technologies (ICICT 2014)

*Keywords:*Primary Language; Linguistic Geometry; Algorithm of Discovery; Artificial Intelligence;mosaic reasoning; proximaty reaosning.

1. Introduction

In our previous papers[2-7] a number of past discoveries were investigated. The goal is to discover and, eventually, implement in software the Algorithm of Discovery utilized by humans throughout the history of humanity. The hypothesis we are developing states that there is a universal algorithm that participates in making all the discoveries in sciences and technology. In a series of papers we investigated a number of past discoveries, especially those in LG. We suggest that all the discoveries are based on the series of thought experiments that manifest themselves via mental "visual streams", the animated movies or plays, reflecting artificial mentally constructed reality and,

* Corresponding author. Tel.: +1-303-717-2110.
E-mail address: boris@stilman-strategies.com

Peer-review under responsibility of organizing committee of the International Conference on Information and Communication Technologies (ICICT 2014)
doi:10.1016/j.procs.2015.02.147

sometimes, past reality[3]. These plays show in the end a solution to the problem simulated in the artificial world. We learned that the Algorithm of Discovery utilizes several classes of visual streams, the so-called Observation, Construction and Validation streams. They can run sequentially or concurrently and exchange information between each other. The streams may initiate other thought experiments, "program" them, and schedule them to be executed later in specific order. The Observation stream, usually, "erases the particulars" of an observed object or a process and generates abstract relations. As a rule, the stream is executed several times with different input. It usually employs the imaginary anthropomorphic entity called the Ghost. As soon as the abstract relations are developed, a construction set and a visual model have been automatically produced.

In addition to the major components of the Algorithm of Discovery utilized for every discovery we are in the search for other components that may not be utilized to its capacity in every one of those. The degree of utilization of each component is tightly connected to the nature of the discovery in hand. It is likely that all the discoveries of the laws of nature include optimization components[7]. As a result, the majority of the thought experiments conducted by the Algorithm of Discovery require some type of optimization. Based on the analysis of multiple case studies[7], it was suggested that this optimization is performed by the imaginary movement via approaching a location in the appropriate imaginary space. We named this type of optimization "proximity reasoning". When utilizing proximity reasoning, the Algorithm of Discovery operates in the space where distances are analogous to the actual distances in the real world. Another type of visual reasoning which is the focus of this paper is the "mosaic reasoning" introduced in[8]. This name is based on the analogy of this type of reasoning to the assembling a mosaic of multiple colorful tiles. The key components of the mosaic reasoning include tiles and aggregates, local and global matching rules, and the unstructured environment. The aggregates may include the generator and the plug-in while the rules may include the transformation and the complementarity matching rules. The preliminary results demonstrate that both proximity and mosaic reasoning are utilized mainly for focusing the Construction stream of the Algorithm of Discovery.

In this paper, we will revisit several results in Linguistic Geometry (LG)[9] employing the Algorithm of Discovery to reveal utilization of mosaic reasoning. We will replay the thought experiments related to the discovery of the LG algorithms for generating the Shortest Trajectories and for the Zone construction while emphasizing mosaic reasoning. We will show that mosaic reasoning may be utilized in several places during the journey of making a discovery. It may be employed at the Observation, Construction or Validation stages of the Algorithm of Discovery. For example, mosaic reasoning could be used by the Observation stream simply because it is the place where the constraints for assembling the mosaic are defined. Note that the visual models produced by the Observation stream do not include information on how to construct the final product (to be discovered), it is the Construction stream's responsibility to employ mosaic reasoning for developing an algorithm for actual construction of this product.

As suggested in[6], the Algorithm of Discovery is executed as a sequence of thought experiments. It seems that the only interface for those experiments consists of the visual streams. As we already mentioned, these streams can be divided into three classes, Observation, Construction and Validation streams. Here is a brief description of each type of visual streams.

Observation stream's main purpose is to generate a set of constraints as an input for the Construction stream. Multiple Observation streams can run in parallel; the stream terminates when the required constraints are established or an analogy is found between the mental objects. The stream erases the particulars of the real world objects and generates abstract relations (represented by visual model) by running the mental process multiple times in different scenarios. As soon as those relations are developed, a *construction set* and a *visual model* have been automatically produced. They are specific to the problem in hand, yet represent an abstract concept such as a class of objects or processes of the investigated problem. The produced visual model is used by the Construction stream as a visual manual.

These prototype models do not include information on how to construct the objects; it is the Construction stream's responsibility to figure that out. When the construction set and the visual model are produced, the Construction stream is initiated. Since the Observation stream established the environment by generating the set of constraints, the Construction streams is ready to build the objects using recognizable components (construction sets

and models). During the construction process some parts are symbolically tagged and the algorithm of construction is developed. It is also represented by the visual stream accompanied by the partial symbolic shell. Both have to be verified and possibly enhanced by the Validation stream.

When the objects are constructed, the Validation stream starts and visually verifies if the constructed object matches the constraints established by the Observation stream. Also, it conducts visual reasoning to verify the correctness of the symbolic tagging produced by the Construction stream. Sometimes, formal validation will require developing its own auxiliary Observation and Construction streams.

The Algorithm of Discovery, specifically, some of its visual streams, use and guide a smart slave the "Ghost" to perform simple tasks during the three stages. This Ghost has very limited skills, knowledge and vision. Observation stream may use the Ghost to get more familiar with the construction set and further explore its features[7]. The Ghost may be used by all the types of visual streams.

As discussed in[8], mosaic reasoning requires finding the proper tiles for assembling a mosaic. Next, the matching rules that control the whole assembly process have to be established. Exploration of the matching rules is the main task of the Observation stream. The matching rules serve as constrains that guide the Algorithm of Discovery in the right direction. Violating the matching rules even once (in a thought experiment) would mess up the final product of the discovery, and the Algorithm of Discovery will terminate this experiment and start from the very beginning. We suggested that some of the matching rules impact the mosaic locally[8] while other rules provide global constraints. Global matching rules must be understood first because they represent the framework of the mosaic construction. Then the global matching rules can be elaborated to the placement of specific tiles, thus, representing the local matching rules. The global matching rules may include, in particular, the requirements of the top-down construction, the global complementarity rules, the statistical rules, etc.[8]. Local matching rules, on the other hand, may include local complementarity rules, interchangeability rules, etc.

2. Revisiting Shortest Trajectory

Here we are going to reveal the components of mosaic reasoning involved in the development of the algorithm for generating all the shortest trajectories in the Abstract Board Game (ABG)[6,7,8,9]. The start, the end, the reachability type, and the length of the shortest trajectories are input parameters of the algorithm to be discovered. The visual model of the shortest trajectory (Fig. 1.- red circles) is basically a group of adjacent tiles that represent a shortest planning path for a piece over the abstract board.

2.1. Mosaic Reasoning at the Observation Stage of Trajectory Generation

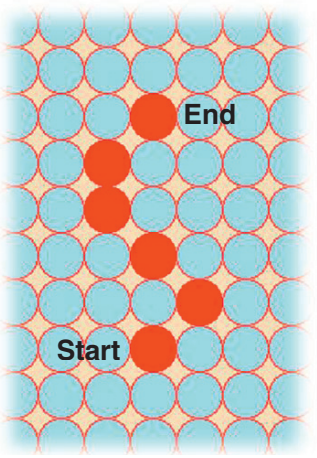

Fig. 1. A visual model of a shortest trajectory.

The Observation stream applies abstraction in order to develop the construction set and the visual model which are inputs for the Construction stream. In our case study, the Observation stream erases the particulars by encapsulating different types of reachability (like different mobility rules for chess pieces)[9]and representing the model of the trajectory as a lane of disk tiles so that every disk tile is reachable from the previous one[6], Fig. 1. The whole mosaic is the network of such lanes (possibly intersecting) with the same start and end. They should be of the same length (the same number of tiles) which represents the length of those shortest trajectories. The stream represented abstract board as a set of small light circles held tightly to each other on the plane, Fig. 1. Each tile of the construction set covers a cell of an abstract board, and reachability is reduced to the visual adjacency. Indeed, eight adjacent cells forming a square shape are reachable from the central cell of this square, Fig. 2. The building blocks of the construction set represent the mosaic environment and should be developed with absolute accuracy. If this accuracy has not been achieved, the whole mosaic will be messed up.

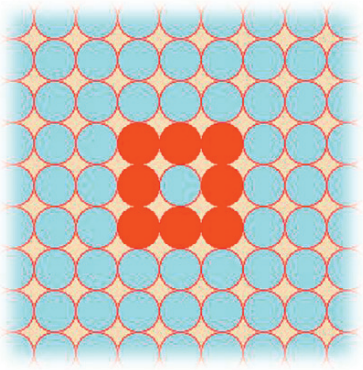

Fig. 2. Eight adjacent cells reachable from the central cell.

For construction of the shortest trajectory, we will consider thought experiments utilized for development of the Grammar of the Shortest Trajectories (Appendix A) in[5, 6, 7]. The Observation stream played the major role in developing means for proximity and mosaic reasoning for this problem. It developed means to measure distances and visualized those distances from the start and the end of the future trajectories for every cell of the abstract board. Those visualizations were named Forward and Backward MAPs while the set of cells where the sum of forward and backward distances is equal to the length of the future trajectories was visualized as the set of SUM (Fig. 3.). By constructing SUM the Observation stream developed the necessary constraints that control the right placement of tiles of all the shortest trajectories as a whole. All the tiles must be within the SUM and no tiles placed outside the SUM could belong to those trajectories. Thus, invention of the SUM aggregate represents development of the global matching rule (the scope rule) that limits the scope of the aggregate for the set of the shortest trajectories by providing precise boundaries for it. This rule clustered and limited the scope of the aggregate to a collection of tiles that belong to the set of all the shortest trajectories as a "pile", without sorting them into separate trajectories. Additionally, the Observation stream introduced another matching rule, the local rule of the mutual adjacency of the tiles that belong to the shortest trajectory which visualized mutual reachability of cells of such trajectory.

2.2. Mosaic Reasoning at the Construction Stage of Trajectory Generation

When the SUM aggregate was initially constructed at the Observation stage, it was not clear how this aggregate of tiles would precisely be used for constructing separate shortest trajectories. The distribution of this aggregate of tiles between the shortest trajectories required a clear definition of the set of MOVE to initiate the process of assembling complete trajectories by transformation. Essentially, the transformation rule here is the algorithm for generating the aggregate of MOVE (Appendix A) while the current tile of the trajectory (just constructed) is the generator. This generator is the tile where the Ghost (entity with eyes) currently stands, Fig. 3.

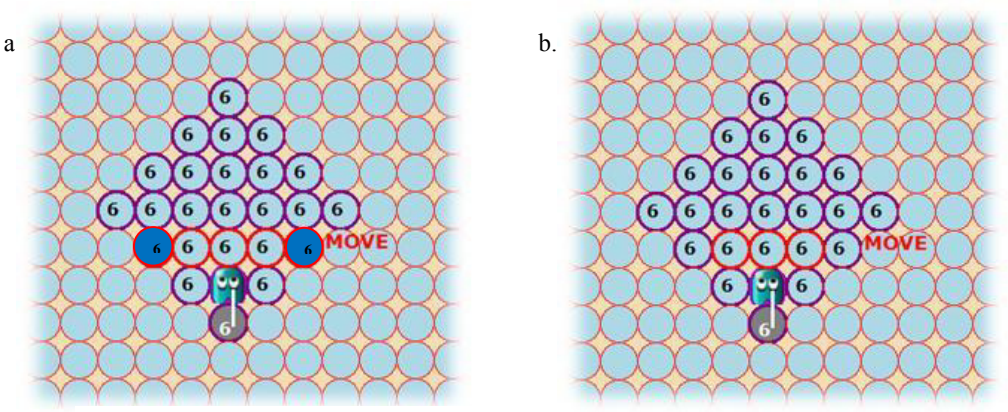

Fig. 3. (a) MOVE is the intersection of only ST_k and SUM; (b) MOVE is the intersection of ST_k, SUM, and ST_1.

During the Construction stage, the first experiment of discovering the algorithm for generating a shortest trajectory was using the intersection of only two aggregates ST_k and SUM for generating MOVE, which resulted in several tiles of the MOVE located too far from x_{k-1}, the previous location of the trajectory, Appendix A and Fig. 3. In other words, the intersection of the two sets produced the aggregate MOVE (Fig. 3.a.) that violated the mutual adjacency rule (the darker blue tiles in Fig. 3.a). In order to fix this problem, in the next experiment, the Construction stream realized that the *k*-th tile along the shortest trajectory must be reachable from the current location, i.e., from the (*k*-1)-st tile x_{k-1}, Fig. 3.b. Thus, the Construction stream concluded that the MOVE must be generated as the intersection of the three sets ST_k, SUM and ST_1 (centred at x_{k-1}), Fig. 4. and Appendix A. Therefore, the local matching rules defined by the Observation stream helped through the process of generating the proper transformation algorithm by the Construction stream.

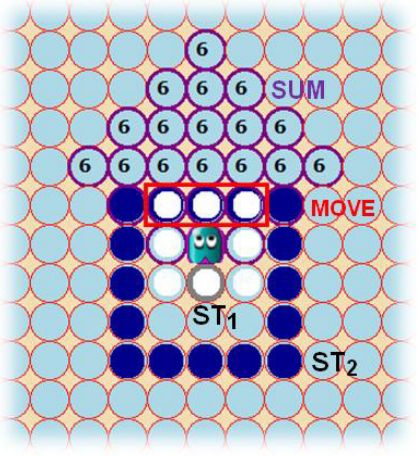

Fig. 4. Construction of the MOVE aggregate.

The global matching rules are related to the whole mosaic. In particular, they determine the area of the major structural component, the large aggregate. In this case study, the global matching rule is the algorithm for constructing the aggregate of SUM. Another global matching rule is the transformation rule based on the algorithm for generating the aggregate MOVE. The local matching rules, on the other hand, are represented here by the complementarity rule. Here, the complementarity rule is the rule of mutual adjacency of the tiles that belong to a

shortest trajectory. The experiments described above have shown that violation of the local rule of mutual adjacency has led to correction of the global transformation rule, the algorithm for constructing and using the MOVE aggregate. Yet another local matching rule is based on the notion of interchangeability in situation where we have several tiles that are interchangeable. Interchangeability means that the visual stream can take one tile (or aggregate) of the mosaic and replace it with another without affecting the structure; this tile is called a "plug-in"[8]. In our case, while constructing a particular trajectory, the Construction stream can pick an arbitrary $Next_i$ component of MOVE as the next cell of this trajectory, Fig. 5. and Appendix A. This means that in this case the plug-in tiles are identical while their locations are different.

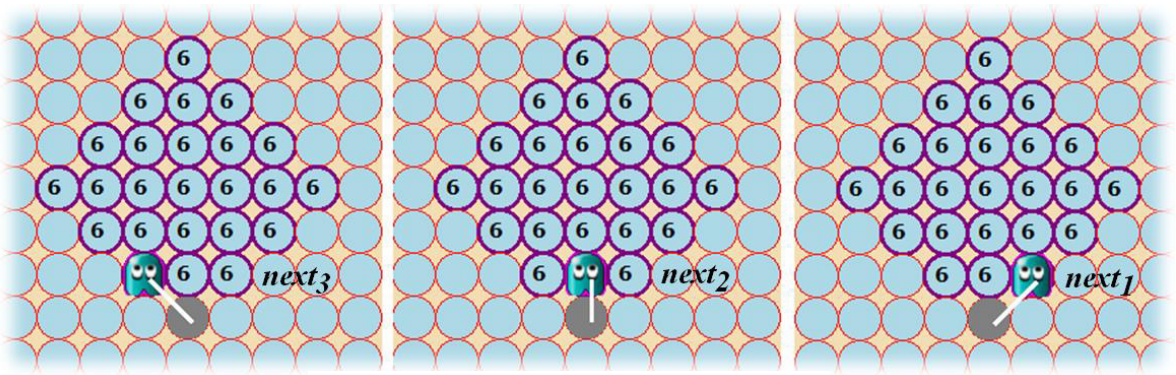

Fig. 5. Interchangeability of the next move.

Different plug-ins certainly change the shape of the chosen shortest trajectory but the whole structure will stand. In other words, plug-ins provide options for choosing the potential *Next* link (tile) from the different choices of tiles in the MOVE aggregate, Fig. 5. Plug-ins played an important role in the discovery of the structure of DNA as it was the key component of the helical structure[8]. However, in the discovery of the algorithm for generating all the shortest trajectories it was a regular branching component of algorithm for generating the bundle of the shortest trajectories (when aggregate MOVE included several tiles). In cases when MOVE included one tile the plug-in reduced to one choice. The branching algorithm was analogous to the depth first search but without the actual search because all the branches are included in the construction of the bundle of the shortest trajectories.

3. Revisiting Zones Generation (Pictorial LG)

Here we are going to reveal the components of mosaic reasoning involved in the development of the algorithm for generating an attack Zone[6,7,8,9]. The main visual stream where mosaic reasoning takes place is the Pictorial LG stream, Fig. 6. The Pictorial LG stream was originated in the development of the algorithm for the computer chess program PIONEER[9], then, the Algorithm of Discovery used it again and again for solving general LG problems[6,7]. The Pictorial LG was a key component and a powerful tool in every discovery related to LG.

3.1. Mosaic Reasoning at the Observation Stage of Zone Generation

The Pictorial LG was developed originally as a set of planning paths for chess pieces. Then, the Algorithm of Discovery erased the particulars by generalizing it and making it applicable not only for the game of chess but also for any Abstract Board Game[9]. The Pictorial LG represents chess pieces by large circles and trajectories by straight lines with small circles. This way it is able to visualize abstract pieces with their reachabilities, their planned movement over the Abstract Board, Fig. 6.

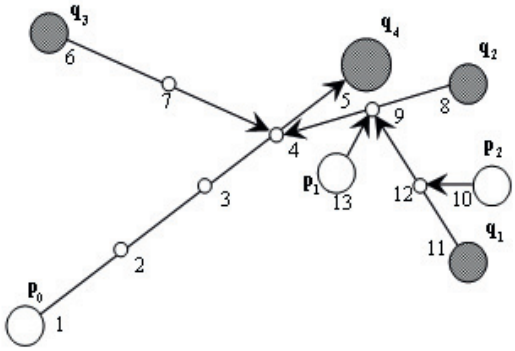

Fig. 6. A Pictorial LG of an attack zone.

The Pictorial LG similarity of trajectories in a Zone caused the Observation stream to pick a trajectory, e.g., the shortest trajectory, as a Zone's generator. While in Pictorial LG all the trajectories look similar like those in Fig. 6., in reality, all the trajectories in a Zone are quite different. They are divided into the main trajectory (1-2-3-4-5), first negation trajectories, (6-7-4 and 8-9-4), second negation (13-9, 11-12-9), and so on. Moreover, they have usually different start and end locations and different length. In addition, they may reflect planning movement of different types of pieces, i.e., they may be based of different reachabilities. All the above differences did not preclude the Algorithm of Discovery from utilizing the same generation algorithm (the essence of the Grammar of Shortest Trajectories, Section 2). Essentially, all those differences were encapsulated in the input parameters of this algorithm. While a trajectory is a Zone's generator it is a plug-in at the same time because there are no two identical trajectories in the Zone.

The Observation stream identified visual transformation for constructing a Zone. This transformation consists of two components:

- changing the length of a line segment, i.e., stretching or shrinking the segment representing a trajectory, the generator, and
- moving this line segment around, i.e., to the proper locations of its start and end.

As we mentioned before, the building blocks of the construction set represent the mosaic environment and should be developed with absolute accuracy. If this accuracy has not been met, the whole mosaic will be messed up. By doing so the Observation stream guaranteed that using Pictorial LG will preserve the structure of the collection of trajectories in the Zone.

3.2. Mosaic Reasoning at the Construction Stage of Zone Generation

The Construction stream utilized transformation (Section 3.1) for developing the Zone generation algorithm (the essence of the Grammar of Zones[9]). It was developed via a thought experiment that used "the principle of timeliness" provided by the Observation stream. This means that the start, the end, and the length of every trajectory in the Zone should be such that the respective piece should always have enough time to participate in the event to happen at the end of this trajectory. By applying this principle consistently the Construction stream developed the "time distribution" matching rules for all the trajectories of the Zone network. In particular, the Construction stream applied abstraction by marking and storing the coordinates of the locations of the ABG where trajectories interconnect[5]. The stream concluded that this would provide required rigidity to the zone and accuracy to the structure of each trajectory.

The time distribution rules allow only legal moves of the opposing pieces' trajectories to be made. The legality of a move here means that the length of the respective trajectory must be less than or equal the amount of time allotted

to it[5]. During the process of constructing the Zone, the Construction stream assigned every trajectory the time allotted to it by following a simple formula shown below (except for the main trajectory). For the main trajectory it simply assigned a value of time equal to the serial number of every stop location (except for the start). For the second and higher negation trajectories it assigned time as follows:

$$\text{Time}(y) - \textit{Length} + 1,$$

Where *Length* is the length of the trajectory and Time(y) is the time allotted to the lower negation at location y while the time for first negation trajectory is simply Time(y) which is the number assigned to the location y of the main trajectory where the first negation is interconnected with[9].

Based on the above analysis, it is clear that the time distribution rule is recursive which requires careful step by step construction. To construct a Zone the Construction stream started with the main trajectory, then it constructed the first negations, then - second negations and so on. The process has to proceed sequentially, i.e., it could not construct, for example the second negation trajectory before the respective first negation trajectory. The time distribution rule insures that the Zone's network of trajectories includes only "active" trajectories.

The global matching rules related to the whole mosaic define the area of the major structural component, the large aggregate. As discussed previously in this paper exploration of the matching rules is the main task of the Observation stream. The matching rules are the absolute constraints that control and guide the Algorithm to the right direction. Violating the matching rules even once (during the construction experiment) would easily mess up the final product of the discovery and the Algorithm of Discovery would terminate and start from scratch. We suggested[8] that some of the matching rules impact the mosaic locally while other rules provide global constraints.

In terms of mosaic reasoning, the time distribution rule is the local complementarity rule. Indeed, the length and the shape of the main trajectory impose constraints on the structure of the first negation trajectories. Recursively the structure and the length of the first negation trajectory imposes constraints on the structure of the second negation trajectories and possible Domination Zones. The number of pieces involved in the Zone and the length of the main trajectory determine the structure of the Zone via multiple repetition of the time distribution rule. The other local complementarity rule is expressed by the adjacency of the trajectories inside the Zone. This means that, by construction, every trajectory (except for the main one) is attached to the location of some other trajectory via its end. None of the trajectories "hang in the air".

Every move in the Zone requires re-computation of time distribution which may result in "disappearance" of some trajectories. Since Zones are dynamic by its nature, the time distribution rule needs to be applied after every move. This rule should to be checked every time when new trajectory is generated. For the "old" trajectories, application of this rule was built into the freeze/unfreeze procedure[9].

4. Conclusion

The process of constructing the shortest trajectory required assembling the trajectory one tile at a time. Mosaic reasoning played a major role as the development of the algorithm for generating the shortest trajectories required a precision so that each tile should be placed in a position that fits its neighbors. This required finding a collection of tiles that certainly fit together, the aggregate of SUM. The set of SUM served as the large aggregate of tiles that satisfy the global matching rule (scope rule) while another global matching rule, the transformation rule represented by the set MOVE guided the placement of the next tiles adjacent of the previous one.

On constructing an attack zone, on the other hand, the structure of the main trajectory guided the rest of the mosaic employing the local complementarity rules, the time distribution and the adjacency of trajectories in the Zone's network.

References

1. J. Von Neumann, *The Computer and the Brain*, Yale U. Press, 1958.
2. B. Stilman, Discovering Components of the Primary Language, *Int Conf. on Intelligent Systems Designs and Applications (ISDA),* Rome, Italy, 2012.
3. B. Stilman, Discovering the Discovery of Linguistic Geometry, *Int. J. of Machine Learning and Cybernetics*, *Springer*, (DOI) 10.1007/s13042-012-0114-8,20 p.,2012. Printed in 2013, Vol. 4, No. 6, pp. 575-594.
4. B. Stilman, Discovering the Discovery of the No-Search Approach, *Int. J. of Machine Learning and Cybernetics*,Springer, (DOI) 10.1007/s13042-012-0127-3,27 p., 2012.
5. B. Stilman, Discovering the Discovery of the Hierarchy of Formal Languages, *Int. J. of Machine Learning and Cybernetics*, Springer, (DOI) 10.1007/s13042-012-0146-0, 25 p., 2012.
6. B. Stilman, Visual Reasoning for Discoveries, *Int. J. of Machine Learning and Cybernetics*, Springer, (DOI): 10.1007/s13042-013-0189-x, 23 p., 2013.
7. B. Stilman, Proximity Reasoning for Discoveries, *Int. J. of Machine Learning and Cybernetics*, Springer, (DOI) 10.1007/s13042-014-0249-x, 31 p., 2014.
8. B. Stilman, Mosaic Reasoning for Discoveries., *J. of Artificial Intelligence and Soft Computing Research*, Vol. 3, No. 3, 26 p., 2013.
9. B. Stilman, *Linguistic Geometry: From search to construction*, Kluwer Academic Publisher (now Springer), 2000.

Appendix A, see[9] for more details.

Grammar of Shortest Trajectories $G_t^{(1)}$

L	Q	Kernel	F_T	F_F
1	Q_1	$S(x,y,l) \rightarrow A(x,y,l)$	*two*	Ø
2_i	Q_2	$A(x,y,l) \rightarrow a(x)A(next_1(x,l), y, f(l))$	*two*	3
3	Q_3	$A(x,y,l) \rightarrow a(y)$	Ø	Ø

$V_T=\{a\}$ $V_N=\{S, A\}$

V_{PR}

$Pred = \{Q_1,Q_2,Q_3\}$,

$Q_1(x, y, l) = (MAP_{X,p}(y) = l)$ $(0 < l < n)$

$Q_2(l) = (l \geq 1)$

$Q_3 = T$

$Var = \{x, y, l\}$

$F = \{f,next_1,...,next_n\}$ $(n = |X|)$,

$f(l) = l - 1$, $D(f) = \mathbf{Z}_+ - \{0\}$, $next_i$ is defined below;

$E=\mathbf{Z}_+ \cup X \cup P$

Parm: $S \rightarrow Var$, $A \rightarrow Var, a \rightarrow \{x\}$

$L = \{1, 3\} \cup two$, $two = \{2_1,2_2,...,2_n\}$

At the beginning of derivation:

$x = x_o$, $y = y_o$, $l = l_o$, $x_o \in X$, $y_o \in X$, $l_o \in \mathbf{Z}_+$, $p \in P$.

Function $next_i$ is defined as follows: $Domain\ (next_i) = X \times \mathbf{Z}_+ \times X^2 \times \mathbf{Z}_+ \times P$;

$MOVE_l(x)$ is the intersection of the following sets: $ST_1(x)$, $ST_k(x_o)$ and SUM, where $k = l_o - l + 1$; $MAP_{x,p}(v)$ is the function of distance between locations $x \in X$ and $v \in X$ for the piece $p \in P$, it is based on reachability relations of p[9];

$SUM=\{v \mid v \in X, MAP_{x_o,p}(v) + MAP_{y_o,p}(v) = l_o\}$,

$ST_k(x)=\{v \mid v \in X, MAP_{x,p}(v)=k\}$;

if $MOVE_l(x)=\{m_1, m_2, \ldots, m_r\} \neq \emptyset$

then

$next_i(x, l) = m_i$ for $i \leq r$;

$next_i(x, l) = m_r$ for $r < i \leq n$,

else

$next_i(x, l) = x$