Acta Scientific COMPUTER SCIENCES

Volume 4 Issue 12 December 2022

Review Article

Seeing and Managing Distributed Worlds with Spatial Grasp Paradigm

Peter Simon Sapaty*

Institute of Mathematical Machines and Systems, National Academy of Sciences, Ukraine

***Corresponding Author:** Peter Simon Sapaty, Institute of Mathematical Machines and Systems, National Academy of Sciences, Ukraine.

Received: January 05, 2022

Published: November 30, 2022

Abstract

This paper presents the latest results on the development of Spatial Grasp (SG) paradigm, allowing it to solve complex problems in holistic and fully distributed way, in two interlinked directions: philosophical-conceptual, and technological-implementational. In the first direction, more details are presented of how SG develops in distributed spaces in the form of waves or even viruses and grasps at the same time solutions of spatial problems. Also how it fundamentally differs from traditional representations of systems and solutions in them as consisting of parts exchanging messages. The SG philosophy is also linked to higher intellectual concepts like perception, awareness, consciousness, and even soul, with mentioning a related discussion organized by the author in the USA. In the other direction, the latest Spatial Grasp Technology (SGT) details are briefed where its Spatial Grasp Language (SGL) interpreters can be networked as spatial computers and cover any terrestrial and celestial spaces. Distributed interpretation mechanisms of some basic SGL constructs are discussed in detail, allowing us to implement spatial SGT functionality without any centralized resources. The paper also provides examples of fully distributed solutions for seeing and evaluating of large distributed phenomena like hurricanes and forest fires, which cannot be observed from any individual points, but can be seen and analyzed as a whole under SGT, just in line with higher-level conceptual orientation and mental-like capability of SG. The developed paradigm allows us to directly express top sense and related holistic method for solving complex problems. also dynamically compose or even create from scratch the implementation environment, providing strictest and shortest way from the problem vision, definition and understanding to the practical solution.

Keywords: System; Distributed System; Parallel And Distributed Computing; Waves; Grasps; Perception; Understanding; Consciousness; Soul; Spatial Grasp Technology; Spatial Grasp Language; Distributed Interpretation; Spatial Vision; Distributed Networking; Pattern Matching

1 Introduction

The aim of this paper is to further deepen and develop the definition and description of the Spatial Grasp (SG) paradigm in the two interlinked directions: philosophical-conceptual and technological-implementational. Having ideologically born more than half century ago and called WAVE in its childhood, it represents a fundamentally different approach to solving problems in large distributed systems and networks, where instead of placement of activities inside them to communicate with each other, ahead of problem solving, it uses a high-level holistic approach. This approach, being above and over these systems, investigates, processes and controls them by dynamically covering and matching their physical or virtual bodies with active recursive code which spreads like waves and viruses, but in difference to them, can also provide any solutions of both local and global problems in fully distributed and parallel mode. SG can also explicitly implement, simulate, or mimic any other existing models and approaches, including creation of any distributed systems with any structures and topologies, putting any needed communicating activities into their nodes, however, considering this as conceptually lower levels solutions, which often appear just automatically on the SG distributed implementation levels. SG matured and strengthened during de-

cades having been prototyped in different countries and tested on numerous applications, including graph and network theory, computer networking, collective robotics, simulation of battlefields, psychology, sociology, security, disaster management, and recently even celestial orbital systems. The results and experience obtained, with numerous international publications, six book including, with seventh one in print, has led the author to write this paper (another book planned too) to further deepen and extend the approach and make more investigations of its power and potential applicability in extended and important applications. The rest of this paper is organized as follows.

Section 2 presents and discusses main ideas of the developed Spatial Grasp (SG) paradigm. It starts with how traditional systems and solutions in them are organized, with all being represented as consisting of parts exchanging messages. Then main principles of SG are explained which provides the world coverage with parallel active code, which has a symbiotic combination of wavelike (even virus-like) and grasp-like spatial activity. It is also shown how this may relate to much higher mental and philosophical concepts like perception, awareness, consciousness, even soul. A discussion is mentioned that took place in the USA in the nineties on differences between the previous SG version called WAVE and the Society of Mind concept and book developed by Prof. Minsky. Section 3 shows how the discussed SG paradigm has resulted in a real technology prototyped and tested on different applications, with main features of its latest version summarized. It equally operates with different worlds including physical, virtual, executive, and any combinations. Recursive Spatial Grasp Language (SGL) is briefed with its rules, different types of spatial variables, and elementary programming examples. Basics of SGL interpretation are also mentioned where networked interpreter copies, potentially in millions to billions, can operate as powerful spatial machines serving SGL scenarios in parallel.

Section 4 reveals some important technical details of the SGL interpretation, not discussed in previous publications, These include parallel advancement in a distributed space which may be asynchronous or synchronized, repetitive propagation as asynchronous or synchronous too, and also using the dynamically created track system for performing different operations on remotely accessed values, by effectively converting it into a hierarchical spatial computer. The latter can also be used in combination of forward, backward, and forward again manners for a further space exploration. Section 5 shows some examples of how to see, understand, and evaluate as a whole of very large distributed phenomena, unobservable from any single point, by their physical or virtual coverage under the SG paradigm. The high level solutions in SGL contain self-spreading and coverage of the regions of interest, like hurricanes and forest fires, for registering their external borders, with similar techniques potentially applicable for much larger, like celestial, cases. It also shows how to work in SGL for finding certain images in arbitrary distributed networks using spatial graph-pattern matching technique. Section 6 concludes the paper by mentioning implementation availability of the latest SGT version, which can be performed even within standard university environments, similar to the previous versions.

2 Spatial grasp (SG) paradigm main ideas

Traditional system representations

The following worlds and meanings are often used when speaking about systems and distributed systems. A system is a group of interacting or interrelated elements that act according to a set of rules to form a unified whole [1]. Distributed system is a system in which components located on networked computers communicate and coordinate their actions by passing messages. The components interact with each other in order to achieve a common goal [2]. A distributed control system (DCS) is a computerized control system for a process or plant usually with many control loops, in which autonomous controllers are distributed throughout the system, but there is no central operator supervisory control. This is in contrast to systems that use centralized controllers, being either discrete ones located at a central control room or within a central computer [3]. The following concepts usually relate to distributed systems too: message-based communications, *distributed agents, and distributed objects* [4], also languages for parallel and distributed computing which provide synchronization constructs whose behavior is defined by a parallel execution model. A concurrent programming language is defined as one which uses the concept of simultaneously executing processes or threads of execution as a means of structuring a program [5].

Spreading, covering, matching and grasping paradigm

The concept discussed here is based on quite different philosophy of dealing with large distributed systems. Instead of representing systems and solutions in them in the form of communicating parts or agents, the Spatial Grasp paradigm developed is orga-

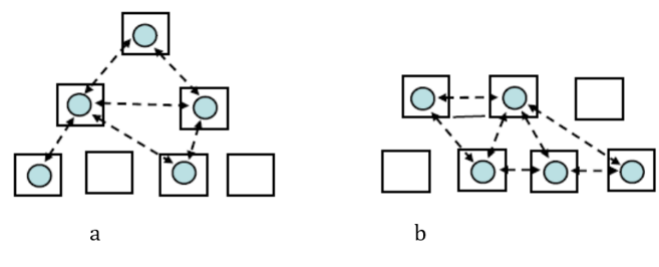

Figure 1: Traditional system representations and solutions in them; (a) hierarchical, (b) distributed.

nizing everything by integral, holistic and parallel self-spreading, self-covering and self-matching distributed spaces, which may be physical, virtual or combined, as symbolically shown in figure 2.

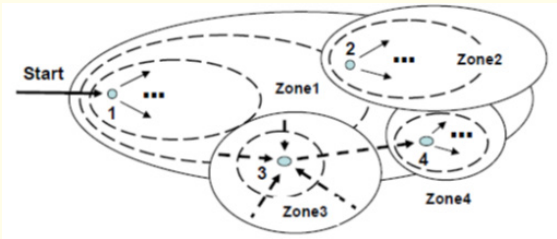

Figure 2: Holistic and parallel world coverage by active recursive code.

Originating in some world point 1 and spreading as Zone 1, it may trigger another space coverage starting in secondary centers (like 2, 3, and 4). In different areas (like Zone 3) complex solutions may be found in both forward and feedback manner, on the results of which another spreading and solution areas may appear (like Zone 4), and so on. Combinations of activation, spreading, and forward-backward problem solving may be arbitrary and by any numbers, depths, hierarchies, and space coverage. Some physical analogies of this paradigm are shown in figure 3 which may include spreading waves or even viruses (Figure 3a), also different grasping capabilities (Figure 3b) where objects or areas seized or grasped could be arbitrary large and of any nature.

Relation of SG to higher mental and philosophical concepts

The paradigm discussed in this paper is philosophically based on much higher and more general concepts than used for the system descriptions mentioned in Subsection 2.1, with some of them

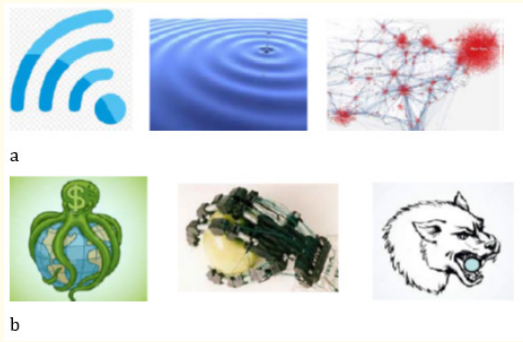

Figure 3: Physical analogies: (a) Spreading waves and viruses; (b) Grasping capabilities.

following. Understanding is a psychological process related to an abstract or physical object, such as a person, situation, or message whereby one is able to use concepts to model that object. Understanding is a relation between the knower and an object of understanding [6]. Also, understanding the problem is often the main part of its solution [7,8]. Perception is the organization, identification, and interpretation of sensory information in order to represent and understand the presented information or environment [9]. Self-Awareness and Mental Perception go even higher [10]. At the highest level is the concept of Consciousness [11], with many theories and fantasies of what it actually means, can it exist outside the head [12] or even does consciousness pervade the Universe [13], also relation of consciousness to space [14,15]. The soul within many religious, philosophical, and mythological traditions is the incorporeal essence of a living being. It is to comprise the mental abilities of a living being: reason, character, feeling, consciousness, memory, perception, thinking, etc. Depending on the philosophical systems, a soul can be either mortal or immortal [16], and such guesses as its possible separation from body [17], when does it enter the human body, or where does it reside [18] are even fantasized too.

WAVE concept against the society of mind

In relation to the ideas discussed above, the author recollects his meeting (about 1996) at Dartmouth College and hot discussions with Prof. Marvin Minsky, one of the fathers of computer science and cofounder of the Artificial Intelligence Laboratory at MIT, also author of the book The Society of Mind [19]. The discussion was about comparison of the previous author's concept called WAVE [20-22] based on the above mentioned ideas and Marvin's portrayal of the human mind as a "society" of tiny components,

or agents, themselves mindless, together like a mosaic, but their frequent interactions were somehow forming of what was called "intellect". His view was close to the traditional systems representations discussed in Subsection 2.1, and the author was presenting the WAVE concept in some sense closer to the notion of "soul". The chat was particularly interesting and hot because Marvin was an ardent atheist, and to convince him that the tech mimics soul rather than body was not easy. Prof. Minsky was even invited to see himself the working WAVE system installed at the University of California at Irvine as public domain, but unfortunately was unable to organize such a visit.

3 Spatial grasp technology basics

We are showing here how the spatial paradigm discussed above has been patented [23] and converted into a real technology prototyped in different countries and tested on numerous applications [24-29]. Within Spatial Grasp Technology (SGT), a high-level scenario for any task to be performed in a distributed world is represented as an active self-evolving pattern rather than a traditional program, sequential or parallel one. This pattern, written in a high-level Spatial Grasp Language (SGL) and expressing top semantics of the problem to be solved, can start from any point of the world. Then it spatially propagates, replicates, modifies, covers and matches the distributed world in a parallel wavelike mode, while echoing the reached control states and data found or obtained for making decisions at higher levels and further space navigation,

The worlds SGL operates with

SGT allows us to directly operate with different world representations: Physical World (PW), considered as continuous and infinite, where each point can be identified and accessed by physical coordinates; Virtual World (VW), which is discrete and consists of nodes and semantic links between them; and Executive world (EW) consisting of active "doers" with communication possibilities between them. Different kinds of combination of these worlds can also be possible within the same formalism, like: Virtual-Physical World (VPW), Virtual-Execution World (VEW), Execution-Physical World (EPW), and Virtual-Execution-Physical World (VEPW) combining all features of the previous cases.

Spatial grasp language (SGL) syntax

SGL top level syntax following.

grasp→ constant | variable | rule [({ grasp,})]

constant → information | matter | custom | special | grasp

variable→ global | heritable | frontal | nodal | environmental

rule→ type | usage | movement | creation | echoing | verification | assignment |

advancement | branching | transference | exchange | timing | qualifying | grasp

An SGL scenario, called grasp, applied in some point of the distributed space, can just be a constant directly providing the result to be associated with this point. It can be a variable whose content, assigned to it previously when staying in this or (remotely) in other space point (as variables may have non-local meaning and coverage), provides the result in the application point too. It can also be a rule (expressing certain action, control, description or context) optionally accompanied with operands separated by comma (if multiple) and embraced in parentheses. These operands can be of any nature and complexity (including arbitrary scenarios themselves) and defined recursively as grasp, i.e. can be constants, variables or any rules with operands (i.e. as grasps again), and so on.

SGL rules

Rules, starting in some world points, can organize navigation of the world sequentially, in parallel or any combinations thereof. They can result in staying in the same application point or can cause movement to other world points with obtained results to be left there, as in the rule's final points. Such results can also be collected, processed, and returned to the rule's starting point, the latter serving as the final one on this rule. The final world points reached after the rule invocation can themselves become starting ones for other rules. The rules, due to recursive language organization, can form arbitrary operational and control infrastructures expressing any sequential, parallel, hierarchical, centralized, localized, mixed and up to fully decentralized and distributed algorithms.

SGL variables

These are: Global variables (most expensive), which can serve any SGL scenarios and be shared by them, also by their different branches; Heritable variables appearing within a scenario step and serving all subsequent, descendent steps; Frontal variables serving and accompanying the scenario evolution, being transferred between subsequent steps; Environmental variables allowing us to access, analyze, and possibly change different features of physical, virtual and executive words during their navigation; and finally,

Nodal variables as a property of the world positions reached by scenarios and shared with other scenarios in same positions.

Elementary SGL programming examples

- add(7,8) -- Finds the sum of two values when staying in some world point and leaves the result there.
- assign(Result, add(7,8)) -- The sum of two values found is assigned to a variable Result which will be staying in the same point.
- move(x_y) -- From the current world point makes physical move to another physical point with given coordinates.
- create(John) -- Creates isolated virtual node John.
- hop(John); create(+father, Peter) -- Extends the existing single node virtual network with new node and relation to it, where John will be treated as father of Peter.
- move(x_y); repeat(shift(dx_dy); TEMPERATURE > 0) -- Starting from world point with proper coordinates, organizes repetitive movement in the chosen direction unless temperature in the reached physical locations remains above zero.
- if(hop(Peter), create(Lilia, Olga, Ann)) -- In case of existence of virtual node Peter, three new isolated virtual nodes with proper names will be created.

4 SGL spatial interpretation

Communicating Interpreters of SGL can be in arbitrary number of copies, say, up to millions and billions, which can be effectively integrated with any existing systems and communications, and their dynamic networks can represent powerful spatial engines capable of solving any problems in terrestrial and celestial environments. Such collective engines can simultaneously execute many cooperative or competitive tasks-scenarios without any central resources or control, as symbolically depicted in figure 4 (SGL interpreters just named U as universal computational and management nodes, which may be stationary or requested and runtime located in proper space points on the demand of SGL scenarios).

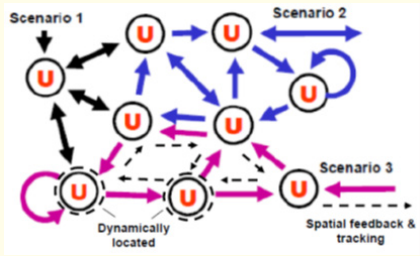

Figure 4: Spatial interpretation of SGL scenarios.

As both backbone and nerve system of the distributed interpreter, its self-optimizing spatial track system provides hierarchical command and control as well as remote data and code access. It also supports spatial variables and merges distributed control states for decisions at different organizational levels in a feedback mode. The track infrastructure is automatically distributed between active components (humans, robots, computers, smartphones, satellites, etc.) during scenario self-spreading in distributed environments.

Examples of some mechanisms of distributed SGL interpretation

Generally, such examples can be better explained on the growing spatial trees, but for compactness, will be shown in a simplified way here.

Advancement: advance(g1,g2,g3) or advance_sync(g1,g2,g3)

Operations g1, g2, g3 are considered to be executed sequentially, one after another, where each new one should be applied in parallel to all positions in space reached by the previous operation. Such scenario can be executed in asynchronous or synchronized manner, as follows.

Asynchronous advancement

In asynchronous solution, each new operation begins developing immediately from any Pi points reached by the previous operation without waiting for its completion in other Pi points, with the rest of the scenario moving in space while omitting the used operations, as in figure 5.

Synchronous advancement

In synchronized case, each new operation can be executed from Pi points reached by the previous operation only after all Pi has

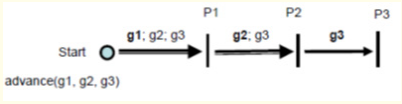

Figure 5: Non-synchronized advancement in space.

been completed, which should be informed to the Start point. After this, the next operation will be launched from all Pi, to which its text should be delivered from the Start, as in figure 6. This mode uses the feedback capability via the SGT tracking mechanism of self-spreading scenarios, which may have different implementations, not necessarily using the whole found paths from the Start to Pi, and even may just be in returning to Start the direct address of Pi nodes, if the latter available.

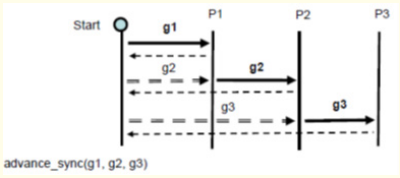

Figure 6: Synchronized advancement in space.

Another solution for the synchronized advancement may be movement in space of the whole scenario text losing each time the utilized operations, similar to the asynchronous case of figure 6, as now in figure 7. By this, after the feedback from full completion of Pi, the Start node will be issuing permission in all Pi to start the next operation, with the rest of the scenario followed by it in space.

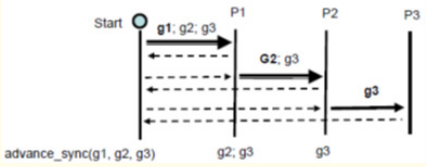

Figure 7: Another variant of synchronized advancement.

Repetitive advancement: repeat(g1);g2 or repeat_sync(g1);g2

We will show here an example of how repeated invocation of operation g1 can be managed in asynchronous and synchronized way, similar to the previous advancement case, after termination of which the next operation g2 should be invoked from all final points reached by the repetition of g1.

Asynchronous repetition

For asynchronous way, the same scenario with repeated g1 with g2 following it will be invoked in all points of Pi reached by the latest g1 copy, and in case it this is impossible, g2 will be immediately invoked from the same points, as in figure 8. In fully asynchronous way, with independent and parallel development is space which may have different features in different directions, it may happen that g1 terminates in different space points after different repetitions, so applications of g2 may happen to be involved not only from Pn as in figure 8, but also potentially from any previous Pi.

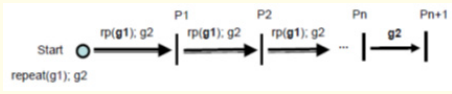

Figure 8: Non-synchronized repeated advancement in space.

Synchronous repetition

For the synchronized repetition in space, each new repetition of g1 can be applied only after full completion in space of all copies of g1, and then to the successfully reached nodes Pi another copy of g1 will be delivered from the Start. In case of its final invocation, to the points Pn from which it failed, operation g2 will be delivered for execution, as in figure 9. This synchronous case may happen to operate differently in comparison with the previous asynchronous one, and failure of g1 in previous Pi will not be followed by g2.

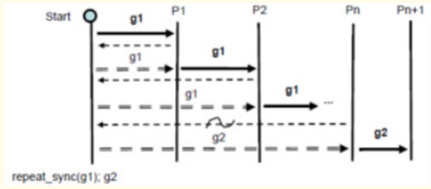

Figure 9: Synchronized repeated advancement in space.

Another solution for the synchronous repetition development may combine with techniques of the previous asynchronous one, where both g1 and g2 always propagate in space between Pi, being left in the reached Pi points for the next step. And from the Start

node only permission to develop g1 further of apply g2 will be issued on the received feedback, without each time delivering of their texts from the Start, as in figure 10.

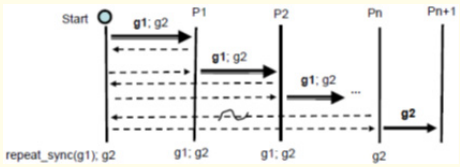

Figure 10: Another variant of repeated advancement.

Feedback-based parallel spatial computation: sum(g1;g2;g3)

This scenario propagates in distributed space stepwise by g1, g2 and g3 operations, similar to Figure 6, with each new one developing from all space points reached by the previous operation, and then summates all values reached by the final operation g3. The figure 11 shows how to use the track tree appeared from the spreading from Start to all final values (which may remote and in large quantity) to make this summation hierarchical and in a feedback manner while preliminary leaving in all forwardly passed Pi points the summation operation. This will be effectively converting the SGT track mechanism into a parallel spatial computer obtaining and returning the final result into the Start position.

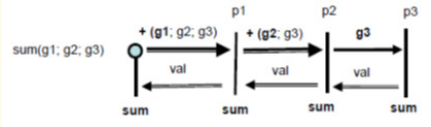

Figure 11: Parallel hierarchical feedback operation.

Combination of feedback computation with forward advancement: maxdest(g1;g2;g3);g4

This scenario, see figure 12, similar to the previous one in Figure 11, propagates in space to the final values obtained by g3 and finds maximum value among them using preliminary left max operation in the Start and passed Pi points. It then delivers the next operation g4 to the point in P3 having this maximum value, using from the

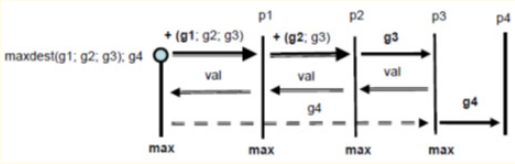

Figure 12: Combining hierarchical feedback operation with further space navigation.

Start the resultant track path to this particular point. So this scenario is working in combination of spatial forward, backward, and then forward again mode, also using the distributed track system as parallel hierarchical computer.

5 Examples of high-level spatial vision and comprehension

Hierarchical coverage in finding borders of a large spatial image

Starting from some point which is definitely inside the region of interest, the following scenario organizes stepwise propagation through the region in different extending directions until the reached points are still classified as being inside the region, as in Figure 13a. And the first points reached in this self-spreading spatial process which are not considered as the region's (say, by a given brightness threshold), are declared as those being on the region's border. Their coordinates will be returned to the Start position in parallel echo process while collected and declared there as the external border of the region.

```
move(Xstart_Ystart); frontal(Threshold = ..., Shift);
Border = unit(
 dx1_dy1, dx2_dy2, ..., dx6_dy6; Shift = VALUE;
 repeat(shift(Shift); if(BRIGHTNESS < Threshold, done(WHERE))));
output(Border)
```

A better and more detailed solution for finding the region's external border can be obtained by the growing tree-like space coverage, where at each step of extending the distance from the Start the tree splits into more branches, as in figure 13b. The finally reached and returned points (which may be much more than in figure 13a) will definitely characterize the region's border with more points

and details.

```
frontal(Start) = Xstart_Ystart, Threshold = ..., Branch = 2, Shift);
move(Start);
Border = unit(
dx1_dy1, dx2_dy2, ..., dx6_dy6; Shift = VALUE; shift(Shift);
 repeat(
 shift(split(find(Start, WHERE, Shift, Branch)));
 if(BRIGHTNESS < Threshold, done(WHERE))));
```

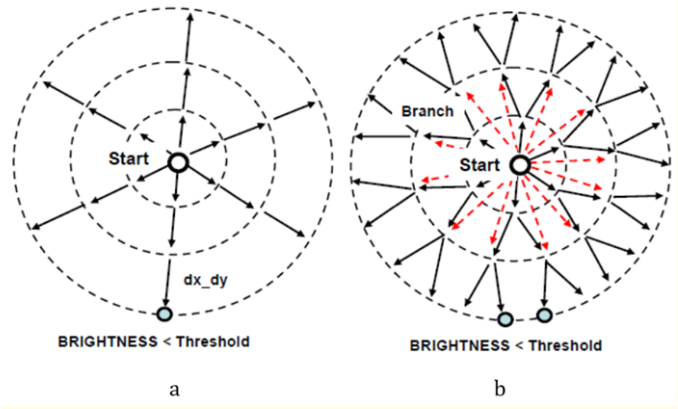

Figure 13: Finding borders of a space region.

```
output(Border)
```

In figure 14a a possible application of the above mentioned tree-like space coverage is shown for finding the external border of a hurricane which can be observed from space as a whole or by its separate points from the ground with obtaining full image as by the above scenario. The latter can also be used in any other applications oriented on seeing and analyzing the extension and shape of much larger spatial formations like, for example, a galactic, as in figure 14b.

Virus-like spatial coverage for finding borders of complex regions

But the solutions shown above may not work well for analyzing very complex regions like, for example, forest fires, as in figure 15, which may have any (including distorted and curved) shapes not so effectively covered by well structured top-down trees. The following very simple solution for such cases is based on the idea of global viruses which can self-replicate and self-spread in very different directions, backward including (using the SGT's resemblance to

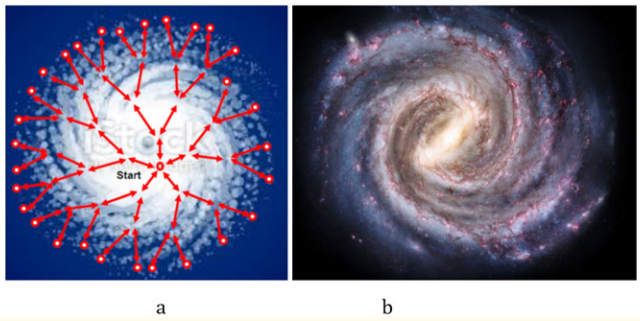

Figure 14: Possible application to seeing and outlining a hurricane (a) or galactic (b).

viruses, as in Section 2). The repeated and free propagation in the solution below does not leave spatial tracks for the feedback use, as in the previous scenarios, but rather individually picks up and sends coordinates of the reached border points directly to the Start position, which finally outputs the collected positions as the border of the region, see figure 15.

```
move(Xstart_Ystart);
frontal(Start = ADDRESS, Limit = steps, Zone_color = ...);
nodal(Border);
sequence(
 free_repeat(Limit)(
 shift(split(find(random(dx_dy), Branch)));
 if(COLOR != Zone_color,
 done(append((hop(Start); Border), WHERE)))),
```

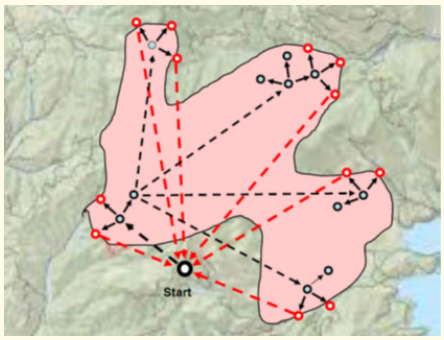

Figure 15: Virus-like investigation and finding borders of a forest fire.

output_unit(Border))

Discovering proper structures in a distributed network

This example relates to spatial seeing of fully distributed networks, like the one shown in 16a, and finding proper structures, or images, in them like in figure 16b, with the latter represented as a search pattern of figure 16c. Many similar tasks and solutions can be found in existing publications on the SGT technology, so we are showing below only the solution for this specific case without detailed explanation. The only comment is that the structure to be found here represents a three node clique where each node has links with other two nodes, but general solutions for any cliques can be easily found in [25-29].

```
hop(all); frontal(X) = NAME;
hop(neighbors); NAME < PRED; append(X, NAME);
hop(neighbors); NAME < PRED; yes(hop(X[1])); append(X, NAME);
hop(neighbors); NAME < PRED; yes(hop_parallel(X[1,2])); append(X, NAME);
output_unit(X)
```

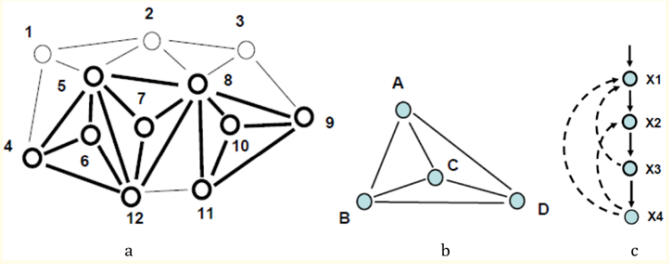

Figure 16: Finding certain graph structures in a distributed network.

The result of this scenario for the network of figure 16a will be three cliques as: (12, 6, 5, 4), (12, 8, 7, 5), (11, 10, 9, 8).

6 Conclusions

This paper described some latest results on the further development of Spatial Grasp (SG) paradigm in two interlinked directions: philosophical-conceptual, and technological-implementational. In the first direction, more details were presented and explained of how SG develops in distributed spaces in the form of waves or even viruses, and how it grasps at the same time complex solutions of spatial problems. Also how this fundamentally differs from traditional representations of distributed systems and solutions in them as consisting of parts exchanging messages. The SG philosophy was also linked with higher intellectual and mental concepts like perception, awareness, consciousness and even soul, with mentioning a related discussion organized by the author in the USA years ago. In the second, technological and implementation direction, the latest Spatial Grasp Technology details were briefed which implement main SG features and capabilities, where Spatial Grasp Language (SGL) interpreters can be arbitrarily networked as spatial computers and cover any terrestrial and celestial spaces. Distributed interpretation mechanisms of some main SGL constructs were revealed and discussed in detail, and for the first time, which allow us to implement powerful spatial functionality without any centralized resources. The paper also provided examples of fully distributed solutions for seeing and evaluating of large distributed phenomena like hurricanes and forest fires, also large networks, which cannot be observed in full from any points, but can be clearly seen and analyzed under SGT providing a sort of their holistic spatial vision and analysis as a whole, thus confirming higher-level conceptual orientation and mental-like capability of the SG paradigm. The latest SGT version can be effectively implemented and used even in traditional university environments, similar to the previous ver-

Bibliography

1. System.
2. Distributed systems.
3. Distributed control system.
4. R Buyya and T Selvi. "Parallel and Distributed Computing". in Mastering Cloud Computing. Elsevier Inc. (2013).
5. List of concurrent and parallel programming languages.
6. Understanding.
7. First, seek to understand the problem.
8. Understanding Your Problem Is Half the Solution (Actually the Most Important Half).
9. Perception.
10. H Kobayashi. "Self-Awareness and Mental Perception". *Journal of Indian Philosophy* 38 (2010): 233-245.
11. Consciousness.

12. F Tonneau. "Consciousness outside the Head". Behavior and Philosophy, The Study of Behavior: Philosophical, Theoretical, and Methodological Challenges 32.1 (2004): 97-123.

13. G Cook. "Does Consciousness Pervade the Universe?" *Scientific American* (2020).

14. D Galland and M Grønning. "Spatial consciousness". In book: The Wiley-Blackwell Encyclopedia of Urban and Regional Studies, Wiley-Blackwell, (2019).

15. C McGinn. "Consciousness and Space". Rutgers University (1997).

16. Soul.

17. A Smith. "Separation of Soul From Body, in Porphyry's Place in the Neoplatonic Tradition" (1974): 20-39.

18. How And When Does The Soul Enter The Human Body and Where Does It Reside, Daaji, Heartfulness.

19. M Minsky. " The Society of Mind". Simon and Schuster, (1988).

20. Sapaty PS. "The WAVE-1: A new ideology and language of distributed processing on graphs and networks". *Computers and Artificial Intelligence* 5 (1987).

21. Sapaty PS. "Logic flow in active data". in VLSI for Artificial Intelligence and Neural Networks (W.R. Moore and J. Delgado-Frias, Eds.), Plenum Press, New York and London, (1991).

22. PS Sapaty. "The WAVE paradigm". Proc. JICSLP'92 Post-Conference Joint Workshop on Distributed and Parallel Implementations of Logic Programming Systems, Washington, D.C., Nov.13-14 (1992).

23. Sapaty PS. "A distributed processing system". European Patent N 0389655, Publ. 10.11.93, European Patent Office. 35.

24. Sapaty PS. "Mobile Processing in Distributed and Open Environments". New York: John Wiley and Sons, (1999): 410.

25. Sapaty PS. "Ruling Distributed Dynamic Worlds". New York: John Wiley and Sons, (2005): 255.

26. Sapaty PS. "Managing Distributed Dynamic Systems with Spatial Grasp Technology". Springer (2017): 284.

27. Sapaty PS. "Holistic Analysis and Management of Distributed Social Systems". Springer (2018): 234.

28. Sapaty PS. "Complexity in International Security: A Holistic Spatial Approach". Emerald Publishing (2019): 160.

29. Sapaty PS. "Symbiosis of Real and Simulated Worlds under Spatial Grasp Technology". Springer, (2021): 251.