From: jeffrey E. <jeevacation@gmail.com>
Sent: Monday, February 19, 2018 12:15 PM
To: Joscha Bach
Subject: Re:

i also see no learning in your system .? . eit=er it instantaneious. . fixed time and fixed time for each . computation t=ough answer partilaly known?

On Mon, Feb 19, 2018 at 6:24 AM, Joscha Bach <[redacted]<mailto[redacted]> > wrote:

As you may have noticed, my whole train of thought on computationalism is b=sed on the rediscovery of intutionist mathematics under the name "com=utation".

ttp://math.andrej.com/=p-content/uploads/2014/03/real-world-realizability.pdf <http://math.andrej.com/wp-content/uploads/2014/03/real-wor=d-realizability.pdf>

The difference between classical math and computation is that classically, = function has a value as soon as it is defined, but in the computational p=radigm, it has to be actually computed, using some generator. This also ap=lies for functions that designate truth. For something to be true in intui=ionist mathematics, you will always have to show the money: you have to de=onstrate that you know how to make a process that can actually perform the=necessary steps.

This has some interesting implication: computation cannot be paradoxical. I= the computational framework, there can be no set of all sets that does no= contain itself. Instead, you'd have to define functions that add and =emove sets from each other, and as a result, you might up with some period=c fluctuation, but not with an illegal state.

Intuitionist math fits together with automata theory. It turns out that the=e is a universal computer, i.e. a function that can itself compute all com=utable functions (Turing completeness). All functions that implement the u=iversal computer can effectively compute the same set of functions, but th=y may differ in how efficiently they can do it. Efficiency relates to comp=tational complexity classes.

The simplest universal computers known are some cellular automata, with Min=ky and Wolfram arguing about who found the shortest one. Boolean algebra i= Turing complete, too, as is the NAND gate, the lambda calculus, and almos= all programming languages. The Church Turing thesis says that all univers=l computers can compute each other, and therefore have the same power.

I suspect that it is possible that the Church Turing thesis is also a physi=al law, i.e. it is impossible to build physical computer that can calculat= more than a Turing machine. However, that conflicts with the traditional =ntuitions of most of physics: that the universe is geometric, i.e. hyperco=putational. The fact that we cannot construct a hypercomputer, not just no= in physics, but also not mathematically (where we take its existence as g=ven when we perform geometry), makes me suspect that perhaps even God cann=t make a true geometric universe.

How can we recover continuous space from discrete computation? Well, spacet=me is the set of all locations that can store information, and the set of =ll trajectories along which this information can flow, as seen from the pe=spective of an observer. We can get such an arrangement from a flat lattic= (i.e. a graph) that is approximately regular and fine grained enough. If =e disturb the lattice structure by adding more links, we get nonlocality (=.e. some

information appears in distant lattice positions), and if we remo=e links, we get spatial superposition (some locations are not dangling, so=we cannot project them to a single coordinate any more, but must project t=em into a region).

On the elementary level, we can define a space by using a set of objects, a=d a bijective function that maps a scalar value to a subset of these objec=s. The easiest way of doing might be to define a typed relationship that o=ders each pair of objects, and differences in the scalar are mapped to the=number of successive links of that relationship type. We can use multiple =elationship types to obtain multiple dimensions, and if we choose the rela=ionships suitably we may also construct operators that relate the dimensio=s to each other via translation, rotation and nesting, so we derive the pr=perties of Euclidean spaces.

To get to relativistic space, we need to first think about how information =ight travel through a lattice. If we just equalize value differentials at =eighboring locations, we will see that the information dissipates quickly =nd won't travel very far. To transmit information over large distances=in a lattice, it must be packaged in a way that preserves the value and a =omentum (in the sense of direction), so we can discern its origin. A good =oy model might be the Game of Life automaton, which operates on a regular =wo dimensional lattice and allows the construction of stable, traveling os=illators (gliders). In Game of life, only the immediate neighbor locations=are involved, so gliders can only travel in very few directions. A more fi=e grained momentum requires that the oscillator occupies a large set of ad=acent lattice locations. SmoothLife is a variant of Game of Life that uses=very large neighborhoods and indeed delivers stable oscillators that can t=avel in arbitrary directions.

I think I have some idea how to extend this toy model towards oscillators w=th variable speed and more than two dimensions. It may also possible to sh=w that there are reasons why stable traveling oscillators can exist in 1d,=2d and 3d but not in 4d, for similar reasons why stable planetary orbits o=ly work in 3d.

To give a brief intution about a traveling oscillator as a wavelet: Think o= a wavelet as two concentric circles, one representing the deviation above=zero, the other one the deviation below zero. They try to equalize, but be=ause the catch up is not immediately, they just switch their value instead= (This is the discretized simplification.) Now displace the inner circle w=th respect to the outer one: the arrangement starts to travel. Making the =attern stable requires distorting the circles, and probably relaxing the d=scretization by increasing the resolution. The frequency of the wavelet os=illation is inversely related to how fast it can travel.

You can also think of a wavelet as a vortex in a traveling liquid. The vort=x is entirely generated by the molecular dynamics within the liquid (which=are our discrete lattice computations), and it does not dissolve because i= is a stable oscillator. The vortex can travel perpendicular to the direct=on of the fluid, which is equivalent to traveling in space. It cannot go a=bitrarily fast: the progression of the liquid defines a lightcone in which=each molecule can influence other molecules, and which limits the travel o= every possible vortex. Also, the faster the vortex moves sideways, the sl=wer it must oscillate, because the both translation and state change depen= on sharing the same underlying computation. It will also have to contract=in the direction of movement to remain stable, and it will be maximally co=tracted at the border of the light cone. (The contraction of a vortex is e=uivalent to giving it a momentum.)

An observer will always have to be implemented as a stable system capable o= state change, i.e. as a system of vortices that interact in such a way th=t they form a multistable oscillator that can travel in unison. From the p=rspective of the observer, time is observed rate of state change in its en=ironment, and it depends on its own rate of change, which in turn depends =n the speed of the observer. This gives rise to relativistic time. Also, t=e observer does not perceive itself as being distorted, but it will normal=ze itself, and instead perceive its environment around itself as being dis=orted. As a result, the observer will always have the impression to travel=exactly in the middle of its light cone. This model seems to recover Loren=z invariance, but with a slight catch: it seems to me that while speed of =ight is constant and there is no preferred frame of reference wrt accelera=ion, the resolution of the universe changes with the speed of the observer= No idea if this is a bug or a feature, or if it will be neutralized by so=ething I cannot see yet before I have a proper simulation.

Obviously, all of the above is just a conjecture. I can make a convincing l=oking animation, and I am confident that many features like simultaneity e=c. will work out, but I don't yet know if a proper numeric simulation =ill indeed work as neatly as I imagine.

> On Feb 18, 2018, at 09:00, jeffrey E. <jeevacation@gmail.com <mailto:jeevacatio=@gmail.com> > wrote:
>
> i want to hear more on your views on projection spaces. .=C2 also feel free to put some more meat on the bones of the thin=ing re lorentz transformations
>
> --
> please note
> The information contained in this communication is
> confidential, may be attorney-client privileged, may
> constitute inside information, and is intended only for
> the use of the addressee. It is the property of
> JEE
> Unauthorized use, disclosure or copying of this
> communication or any part thereof is strictly prohibited
> and may be unlawful. If you have received this
> communication in error, please notify us immediately by
> return e-mail or by e-mail to <mailto:jeevacation@gmail.com> , and
> destroy this communication and all copies thereof,
> including all attachments. copyright -all rights reserved

--

=A0 please note

The information contained i= this communication is confidential, may be attorney-client privileged,=may constitute inside information, and is intended only for the use =f the addressee. It is the property of JEE Unauthorized use, disclos=re or copying of this communication or any part thereof is strictly pro=ibited and may be unlawful. If you have received this communication =n error, please notify us immediately by return e-mail or by e-mail to =a href="mailto:jeevacation@gmail.com" target="_blank">jeevacation@gmai=.com, and destroy this communication and all copies thereof, inc=uding all attachments. copyright -all rights reserved

--001a114ca25093257b05658fa510-- conversation-id 14861 date-last-viewed 0 date-received 1519042516 flags 8590195713 gmail-label-ids 7 6 remote-id 796253