From the vision of Norbert Wiener to polytope and ellipsoid uncertainty, topological data analysis, topological quantum computing, fuzzy techniques, and color optical computing

Michael Beer[1], Vladik Kreinovich[2]m Olga Kosheleva[2], Victor Timchenko[3], and Yuriy P. Kondratenko[4],

[1]Leibniz University of Hannover, Germany
[2]University of Texas at El Paso, USA
[3]Admiral Makarov National University of Shipbuilding, Ukraine
[4]Petro Mohyla Black Sea National University, Ukraine
contact email vladik@utep.edu

Abstract

Experiments have shown that as we get closer and closer to an object, our perception goes through several distinct stages. In his groundbreaking book "Cybernetics", Norbert Wiener provided a theoretical explanation of this phenomenon, explanation based on the notion of invariance. In this paper, we show that similar ideas can explain empirical success of many uncertainty-related : use of polytopes and ellipsoids in describing uncertainty, topological data processing, topological quantum computing, fuzzy techniques, and color optical computing.

Keywords: Mobile devices, invariance, ellipsoid uncertainty, polytopes, topological data processing, topological quantum computing, fuzzy techniques, color optical computing.

1 Formulation of the problem

The main challenge that we deal with in this paper. One of the main challenges of mobile systems is that they are more limited in resources than stationary systems. As a result, mobile systems can only afford sensors and computational devices with smaller accuracy and larger uncertainty. An important challenge is: how to deal with this additional uncertainty? Which techniques should we use?

What we do in this paper. In this paper, we show in many cases, the answer to this uncertainty-related challenge can be found by using ideas of Norbert Wiener – the father of Cybernetics. We show that these ideas can explain the empirical success of many current uncertainty-related techniques in different areas ranging from ellipsoid approach to uncertainty to topological data analysis and topological quantum computing to techniques and to color optical computing.

The structure of this paper. We start with describing Wiener's vision in Section 2. In Section 3, we show that this vision can help explain the efficiency of polytopes and ellipsoids in describing set-theoretic uncertainty. In Section 4, we use this vision to explain the efficiency of topological data analysis and topological quantum computing. In Section 5, we use this vision to explain the efficiency of fuzzy logic. In Section 6, we use this vision to explain the efficiency of color optical computing. The final Section 7 contains the proofs.

2 Wiener's vision

What motivated Wiener. In the early 1940s, we already had machines that performed – or at least helped to perform – many manual operations. Machines helped manufacture goods, excavators helped to dig, elevators helped to move people up and down the building, cars, trains, and planes helped move people around, machine like tractors helped in agriculture, mechanical calculators helped with simple computations – e.g., in accounting, etc. As a result of these machines, many difficult and boring jobs – such as digging earth for building's foundations – became partly or fully automated. It became desirable to automate – partly or fully – many other boring jobs – e.g., more complex mechanical jobs, more complex computations needed in accounting and finance, etc.

In many cases in the past, main ideas for automation came from observing natural systems that perform similar jobs. For example, to find out how to make a flying machine, people looked at birds. Birds fly by using wings and tails – and people succeeded in building a flying machine by appropriately designing wings and tails. Of course, airplanes are different from birds: e.g., in contrast to the birds who flap their wings, the airplanes' wings remain stable. However, some main ideas indeed came from birds.

So, to help automate other features – such as vision and reasoning – researchers started analyzing how people do it. To Norbert Wiener, the need for such analysis became clear when his psychology colleague accidentally saw Wiener's draft design of a vision device and noticed that it looked almost exactly like the biological neural network responsible for human vision – it was so much alike that at first, he was under the impression that Wiener became interested in the mechanics of human vision.

This episode – and the knowledge of many other episodes when a biological analogy helped to improve an engineering design – made Wiener realize that there are many common general features between biological systems (ranging from individual creatures to groups and societies) and machines. Wiener was not the first to use biological and social examples to design and improve machines

– after all, airplanes appeared several decades before that – but he was the first to start a systematic study of all such examples and all related general features. He called such general studies *Cybernetics* – from the Greek word *kybernetes* meaning a person who controls (literally who controls a ship). During the Second World War, Wiener's main efforts were focused on using automation to help with the war efforts. Specifically, he focused on the two related important problems:

- how to help anti-aircraft gunners to shoot down enemy airplanes, and
- how to help pilots to avoid being shot down.

His war activities were, of course, classified. After the war ended, most of his results were declassified, so we transformed his wartime research report into a book – called *Cybernetics* – that was published in 1948 [35].

Wiener's vision. In addition to the airplane-related problems, the book also dealt with the problem of creating devices that would emulate human vision in processing visual information – the problem that started his cybernetic studies. To solve this problem, he looked at human vision from an interesting new angle.

Specifically, he took into account that when we view an object under perfect visual conditions – perfect lighting, close location, clear no-fog atmosphere – our vision systems enables us to see numerous small details, and we can perfectly identify this object. However, when the conditions are not perfect – when light is dim, and/or when there is some fog, and/or when the object is far away – our vision is not perfect: based on what we see, we cannot distinguish several somewhat different objects. For example, from a distance, we may be able to see the person's height and beard, but we may not be able to clearly see his face so we may not conclude which of our friends and colleagues we see.

As the visual conditions become better – e.g., when we move closer and closer to the object or when we light the scene better – we start to see see more and more details – until finally we get a perfect image.

Wiener was a mathematician by training. In mathematics, the main objects are sets and functions (transformations), so this is what he looked for when solving practical problems. In vision, we want to reconstruct the 2-D object based on what we see – for example, the shape of this object, which is a set in the 2-D space. In these mathematical terms, the uncertainty with which we see a faraway object means, in particular, that there are some transformations T of the 2-D space that do not affect what we see: the original object S and the transformed object $T(S)$ look to us exactly the same. For example, at a distance, we cannot recognize the size of the object: we cannot tell whether what we see is a taller man who is further way or a shorter man who is nearer. In this case, the transformation is scaling $x \mapsto \lambda \cdot x$ for some $\lambda > 0$.

In mathematics, when some transformation does not change the result, this is known as *invariance.* Invariances are ubiquitous, and mathematicians have studies them for quite some time. Already in the 19th century, mathematicians noticed that for each situation, the class G of all corresponding invariances has a natural structure:

- If any object S is indistinguishable from its transformed version $T(S)$, this means that, vice versa, each object $S' = T(S)$ is indistinguishable from the result $S = T^{-1}(S')$ of applying the inverse transformation T^{-1}.

- Similarly, if transformations T and T' are invariances, then an object S is indistinguishable from $T(S)$, and $S' = T(S)$ is indistinguishable from the object $T'(S') = T'(T(S))$. Since S cannot be distinguished from $T(S)$ and $T(S)$ cannot be distinguished from $T'(T(S))$, this means that S cannot be distinguished from $T'(T(S))$. Thus, for each two invariance T and T', their composition $T' \circ T : S \mapsto T'(T(S))$ is also an invariance.

So, the class of all transformations that we cannot see must contain:

- for each transformation T, its inverse T^{-1}, and

- for every two transformations T and T', their composition $T' \circ T$.

In mathematics, such classes are known as *transformation groups*.

So, Wiener started to study transformation groups corresponding to human vision. For this study, he used the results of numerous human vision studies. These studies showed that as the object gets closer, we pass through several distinct stages. Wiener managed to transform the imprecise natural-language description of these stages into exact language of transformation groups. This is what he found out:

- When the noise level is high (or when the object is far away), all we see is a blurb, we do see any details. This corresponds to the group of all possible transformations, continuous or not.

- As noise level decreases, we start distinguishing whether the object is whole or has a hole (or several holes) inside, but we still cannot see any other details.

- As the object gets closer, we get the group of all projective transformation, i.e., transformation corresponding to a projection from one plane to another.

- Later, we get the group of all affine transformations – i.e., linear transformations

  $$x_i \mapsto \sum_j a_{ij} \cdot x_j + a_i,$$

  for all possible values of a_{ij} and a_i. For example, we can distinguish between an angular shape and a smooth shape, but we cannot yet distinguish a circle from an ellipse or a square from a parallelogram.

- Further on, we get the group of all scalings, shifts, and rotations – this is exactly the case that we mentioned earlier, when we can detect the shape but not the size.

- Finally, when we get close enough to be able to use our binocular (2-eyed) vision to detect the depth, we get the exact image – i.e., image modulo shifts (and maybe small rotations).

Wiener not only provided precise mathematical description of the known vision stages, he also made an interesting hypothesis. Since we humans are a product of billions of years of improving evolution, Wiener proposed a natural hypothesis that only such transformation groups are mathematically possible – if there was some other possible intermediate transformation group, it would have been utilized by the survival-of-the-fittest evolution.

Wiener could not prove his mathematical hypothesis, and at first, not one tried to prove it – Wiener's argument was not taken seriously by other mathematicians: they have learned many time that justifying mathematical results based on what we see is often misleading, However, after a decade during which cybernetics became more and more successful, mathematicians decided to look into this – and, somewhat surprisingly, this hypotheses turned out to be true; see, e.g., [8, 32] (see also [23] for an important 1D analogue).

Comment. By the way, nowadays, while the very word *Cybernetics* is not used as frequently as it used to be, the main directions of today's AI follow exactly the idea of emulating how we human reason. Different directions come from the fact that we can simulate human reasoning on different levels:

- We can actually simulate the resulting reasoning; this corresponds to the logical approach to AI, in particular, fuzzy approach (see below).
- We can simulate what is happening in the cells (neurons) that form our brain when we perform reasoning – this leads to (artificial) *neural networks*, which are, at present, the main AI technique; see, e.g., [7].
- Finally, we can also simulate the evolution that has led both to the brain structure and related techiques; this is known as *evolutionary computations*; see, e.g., [5].

Our opinion is that, while now, the focus is almost exclusively on neural networks, utilizing all three levels is a way ti make AI even more successful – and several papers by Kelly Cohen and his groups show that this combination can indeed lead to many successful applications; see, e.g., [31].

3 Wiener's vision helps explain the empirical success of polytope and ellipsoid uncertainty

Analysis of the problem. Uncertainty means that we do not know the exact values of the parameters $x_1, \ldots, x_n$ describing the object of interest. Instead of a single tuple $x = (x_1, \ldots, x_n)$ of values, we have a set of possible tuples. There are infinitely many possible sets, and we need infinitely many parameters to describe them – while in a computer, we can only store finitely many parameters.

So, to describe the corresponding uncertainty in a computer, we need to use a finite-parametric family of approximating sets.

When sets can be determined only modulo some transformation group, it is reasonable to select a family that is invariant with respect to all the corresponding transformations.

Specifics of mobile devices. For mobile devices, whose ability to store and process information is limited, it is important to limit the number of stored parameters as much as possible.

Why ellipsoids. For the groups of affine and projective transformations, the invariant family of bounded sets with the smallest number of parameters is the family of all ellipsoids; see, e.g., [6, 15, 17, 29, 30]. Ellipsoids have indeed been effectively used to describe uncertainty; see, e.g., [4]. The above result explains this empirical success.

Comments.

- For the group of all scalings, shifts, and rotations, the invariant family of bounded sets with the smallest number of parameters is the family of spheres – which are particular case of ellipsoids.

- Often, we need to combine two or more different estimates. When each estimate results in the conclusion that the actual tuple lies in an ellipsoid, the overall conclusion is that the actual tuple belongs to the intersection of these ellipsoids. Such intersections – called *ellitopes* – have also been successfully used in the analysis of uncertainty; see, e.g., [1].

- In addition to describing the set of possible values, it is desirable to also describe the probability of different values within this set. It turns out that the invariance ideas can also help to describe the corresponding probability distributions; see, e.g., [30].

Why polytopes. Sometimes, our information is not sufficient to limit the set of possible tuples to a bounded set, so we need to consider possibly unbounded sets. In this case, the family with the smallest number of parameters is the family of all half-spaces – i.e., sets described by a linear inequality

$$a_1 \cdot x_1 + \ldots + a_n \cdot x_n + a_0 \geq 0.$$

When we have several estimates, each of which leads to a half-space, we can then conclude that the actual tuple x belongs to the intersection of these half-spaces. Such intersections are nothing else but convex polytopes. An important practically useful case is *boxes*, when each half-space has the form $x_i \geq \underline{a}_i$ or $x_i \leq \overline{a}_i$ for some i. In this case, a bounded intersection consist of all the tuples $x = (x_1, \ldots, x_n)$ for which, for each i, the value x_i belongs to the interval $[\underline{a}_i, \overline{a}_i]$. Because of this, techniques processing boxes are known as *interval techniques*; see, e.g., [9, 16, 20, 22].

4 Wiener's vision helps explain the empirical success of topological data processing and topological quantum computing

From vision to data processing. Wiener's analysis of sets and transformations was related to vision, but similar ideas can be used in data processing. Specifically, when we examine a new phenomenon, first, researchers try to analyze the relation between different quantities $x_1, \ldots, x_n$ that describe the state of the corresponding system. For this purpose, we measure these quantities in different situations. The set of all measured tuples $(x_1, \ldots, x_n)$ forms a set – a subset of the n-dimensional space of all possible tuples.

When the noise level is low, and measurements are very accurate, then the exact shape of this set provides us with a good understanding of the desired relations. For example, when $n = 2$ and the set of possible tuples (x_1, x_2) is a straight line, this means that we have a linear relation between x_1 and x_2. However, often, the noise level is high, so – similarly to the case of vision – based on the observed set, we cannot tell whether this is the actual set of tuples or whether what we observe was obtained from the actual set by some transformation. When the noise level is very high, then, similarly to the case of vision under high noise, we can have all possible continuous transformations. In this case, the only conclusions that we can make based on the set of observed tuples are conclusion that relate to the set's *topology* – i.e., characteristics of this set that are invariant with respect to all possible one-to-one continuous transformations.

The use of such characteristics is known as *topological data processing*; see, e.g., [3, 10, 27]. For example, if the observed pairs (x_1, x_2) form a closed curve, this means that we have a hysteresis phenomenon. This technique has many practical applications – and the above arguments (which are similar to original Wiener's arguments about vision) explain why.

Topological quantum computing. It is known that many problems can be solved much faster when we can use quantum phenomena to process data; see, e.g., [25]. This use is known as *quantum computing.* With quantum computing, we can find an element with a given property in an unsorted n-element array in time $\sqrt{n}$, while without quantum phenomena, it is not possible to do it without performing at least n computational steps (to look at all n elements of the array). An even more spectacular example is coding: quantum computers potentially enable us to decode, in reasonable time, all the messages that are currently encoded by RSA or similar encoding schemes – schemes that are currently used to provide privacy on the Internet.

All this is theoretically possible, but it is not easy to implement quantum computing techniques. One of the main reasons is that to do that, we need to maintain the quantum states for a significant amount of time, while even a small level of noise changes the states very fast. One of the promising ways to solve this problem is to use techniques that rely not on the exact value of the state, but on the *topology* of a multi-particle state, topology that, by definition, does not change after continuous perturbations. This idea is known as *topological quantum computing*; see, e.g., [28] – and it follows the same pattern as topological data processing.

5 Wiener's vision helps explain the empirical success of fuzzy techniques

Why do we need fuzzy techniques. In the early 1960s, Professor Lotfi Zadeh – who was, at that time, one of the world's leading specialists in control – noticed that in many practical situations, skilled humans control the corresponding system better than the supposedly optimal automatic controllers. Of course, one cannot be better than optimal, so the problem is that the models used to design the automatic controllers are not fully adequate: there is some knowledge that skilled human experts have, knowledge that is not yet incorporated in the model. When he asked the experts, they confirmed that there is some additional knowledge that it not yet used – but the problem was that they described this additional knowledge not in precise easy-to-implement terms, but by using imprecise ("fuzzy") words from natural language liks "small". For such words, it was not clear how to incorporate this knowledge.

So Zadeh realized that there is a need for techniques that would help translate such knowledge into precise computer-understandable terms. He called such techniques *fuzzy*, and he developed an example of such techniques. Later, several modifications of his fuzzy techniques have been developed and successfully used; see, e.g., [2, 11, 21, 24, 26, 36].

Main idea behind Zadeh's fuzzy techniques. One of the main reasons why natural-language properties like "small" are difficult to formalize is that, in contrast to precise properties like "greater than 5" – which are, for any given number, either true or false – for some numbers, experts cannot say with absolute accuracy whether a given value is small. When asked, they may say that it is somewhat small, or small to some degree, etc. In other words, such properties are often only satisfied to some degree. So, to describe such properties, we need to be able to describe degrees of confidence intermediate between absolutely false and absolutely true.

In a computer, "true" is usually represented by 1 and "false" by 0. From this viewpoint, it makes perfect sense to use real numbers between 0 and 1 to describe such degrees. A natural way to elicit such a degree is to ask the expert to estimate, on a scale from 0 to 1, to what extent the given number satisfies this property – e.g., to what extent the given value x is small. People can do that: this is how we evaluate the quality of a service, this is how student evaluate their professors – the scale may be different, o to 10 or 0 to 4, but the principle is exactly the same.

So, to describe a property like "small" in computer-understandable terms, we ask an expert to describe, for many possible values x, the degree $m(x)$ to which the value x satisfies the given property – e.g., to what extent x is small. This way, we have a function $m(x)$ from real numbers to the interval $[0,1]$. Zadeh called such functions *membership functions*, or, alternatively, *fuzzy sets.*

Need for "and"- and "or"-operations. In many practical situations, expert rules contain two or more conditions. For example, a medical doctor may say "If a patient has low blood pressure *and* a high fever, then do the following". To describe such rules in computer-understandable terms, we need to be able, for each pair of numbers (x,y), to describe the expert's degree of certainty that blood pressure x is low and temperature y is high. We may have three or more such conditions, in which case we need a degree for each tuple of three or more numbers. While we can elicit degree for many values x – be it 10 or even more – it is not realistic to ask the exapert $10^3 = 1000$ questions about all possible triples of 10 numbers.

Since we cannot directly elicit all such degrees from an expert, we therefore need to estimate such degrees based on whatever information we already have elicited. Specifically, we need to estimate the degree to which a statement $A \,\&\, B$ is true – based on the known degrees a and b describing to what extent A and B are true.

The algorithm $f_\&(a,b)$ for such estimation is called an *"and"-operation* or, for historical reasons, a *t-norm.* Similarly, we need an algorithm that would estimate the degree of an "or"-statement $A \vee B$ based on$a$ and b. Such algorithms are called *"or"-operations* or *t-conorms.*

Which "and"- and "or"-operations should we select? There are many different functions, which of them shall we choose? There are some reasonable restrictions on functions that can be "and"-operations.

- First, it must coincide with the usual 2-valued "and"-operation when each of the statements is either true or false, i.e., we should have $f_\&(0,0) = f_\&(0,1) = f_\&(1,0) = 0$ and $f_\&(1,1) = 1$.
- Second, the degree of confidence in $A \,\&\, B$ cannot be larger than the degree of confidence in each of the statements, so we must have $f_\&(a,b) \le a$ and $f_\&(a,b) \le b$.
- Third, small changes in the degrees a and b should lead to small changes in the estimate $f_\&(a,b)$. In mathematical terms, this means that the function $f_\&(a,b)$ must be continuous.

Finally – in line with the general theme of this paper – we should have some invariance. This is not a requirement usually imposed on a t-norm, so let us analyze what will happen if we impose it. Specifically, there are no guidances on how to assign degree, the only requirement is that smaller confidence should be described by a smaller degree. If values a, b, $\ldots$, assigned by one expert satisfy this requirement, then values $a' = f(a)$, $b' = f(b)$, $\ldots$, for some 1-1 monotonic function from $[0,1]$ to $[0,1]$ also satisfy this condition. So, we can have different scales, with 1-1 functions $f(a)$ and $f^{-1}(a)$ transforming from one scale to another and back. A natural invariance condition is that the result of applying the "and"-operation should not depend on the scale. In other words, the following two procedures should lead to the exact same result:

- we can simply apply the "and"-operation to the given values, resulting in the value $c = f_\&(a, b)$;
- alternatively, we can first transform the given degrees into a new scale, by computing $a' = f(a)$ and $b' = f(b)$, then apply the "and"-operation to the new values, resulting in $c' = f_\&(a', b') = f_\&(f(a), f(b))$, and then move this value back to the old scale, returning $f^{-1}(c') = f_\&(f(a), f(b))$.

Similarly for an "or"-operation $f_\vee(a, b)$:

- First, it must coincide with the usual "or"-operations when each of the statements is true or false, i.e., we should have $f_\vee(0, 0) = 0$ and $f_\vee(0, 1) = f_\vee(1, 0) = f_\vee(1, 1) = 1$.
- Second, the degree of confidence in $A \vee B$ cannot be smaller than the degree of confidence in each of the statements, so we must have $a \le f_\vee(a, b)$ and $b \le f_\vee(a, b)$.
- Third, the function $f_\vee(a, b)$ must be continuous.

Finally, the function must be invariant, i.e., we must have

$$f_\vee(a, b) = f^{-1}(f_\vee(f(a), f(b))$$

for any monotonic 1-1 function $f(a)$ from $[0, 1]$ to $[0, 1]$.

It turns out that these conditions uniquely determine the "and"- and "or"-operations:

Definition 1.

- *We say that a function $F(a, b)$ is* scale-invariant *if we have $F(a, b) = f^{-1}(F(f(a), f(b))$ for any monotonic 1-1 function $f(a)$ from $[0, 1]$ to $[0, 1]$.*
- *We say that a continuous scale-invariant function $f_\&(a, b) : [0, 1] \times [0, 1] \mapsto [0, 1]$ is a* scale-invariant "and"-operation *if $f_\&(0, 0) = f_\&(0, 1) = f_\&(1, 0) = 0$, $f_\&(1, 1) = 1$, and for all a and b, we have $f_\&(a, b) \le a$ and $f_\&(a, b) \le b$.*
- *We say that a continuous scale-invariant function $f_\vee(a, b) : [0, 1] \times [0, 1] \mapsto [0, 1]$ is a* scale-invariant "or"-operation *if $f_\vee(0, 0) = 0$ $f_\vee(0, 1) = f_\vee(1, 0) = f_\vee(1, 1) = 1$, and for all a and b, we have $a \le f_\vee(a, b)$ and $b \le f_\vee(a, b)$.*

Proposition 1.

- *There exists only one scale-invariant "and"-operation $f_\&(a, b) = \min(a, b)$.*

- *There exists only one scale-invariant "or"-operation* $f_\vee(a,b) = \max(a,b)$.

Comments.

- As we have mentioned in Section 1, for reader's convenience, all proofs are placed in a special Proofs section – Section 7.
- This result explains why in many cases, out of many possible "and"- and "or"-operations, the use of minimum and maximum leads to successful applications.

6 Wiener's vision helps explain the empirical success of color optical computing

Why optical computing. As we have mentioned, light travels faster than anything else, so if we use light, we have the fastest possible communications. Also, light consists of many photons that operate independently, so we have a potential for parallel computing – which also speeds up computations. Because of these two reasons, light-based "optical" computing has been successfully used in many applications; see, e.g., [18].

Why color optical computing. Traditional optical computing techniques mostly use only the intensity of the light. However, we can encode more information if we use light signals of different colors. For example, since all human vision is based on the combination of three basic colors – red (R), green (G), and blue (B), it makes sense to use signals of these three colors.

Let us take into account specifics of mobile devices. The main objective of this paper – that we mentioned at the beginning of this paper – is to help mobile devices. The main specifics of mobile devices is that they have limited access to energy. Thus, for mobile uses, we need to develop techniques that use less energy.

In optical computing and optical communications, a significant amount of energy needs to be used to maintain the signal's intensity. Indeed, any signal degrades with time, so we need to periodically amplify it – this is true for long-distance communications, this is true for computations. To minimize this use of energy, it is desirable to come up with methods that do not require such maintenance, i.e., that work when even each of the three color components of the signals decreases.

Let us describe this requirement in precise terms. Each color signal can be described by a triple of non-negative intensities $(x_{\mathrm{R}}, x_{\mathrm{G}}, x_{\mathrm{B}})$ of the three different color components. The above requirement is that the system should work the same way both for the original signal and for the possibly degraded signal, when each of the components is decreased by a certain factor $0 < d \le 1$. In general, for different colors, the degrading factors may be different. So, we arrive at the following definition.

Definition 2. *We say that the triples* $(x'_{\rm R}, x'_{\rm G}, x'_{\rm B})$ *and* $(x''_{\rm R}, x''_{\rm G}, x''_{\rm B})$ *of non-negative numbers are* equivalent *if there exists a triple* $(x_{\rm R}, x_{\rm G}, x_{\rm B})$ *and coefficients* $d'_{\rm R}, d'_{\rm G}, d'_{\rm B}, d''_{\rm R}, d''_{\rm G}, d''_{\rm B} \in (0,1]$ *for which, for each color* i, $x'_i = d'_i \cdot x_i$ *and* $x''_i = d''_i \cdot x_i$.

It is easy to see that this is indeed the equivalence relation. The corresponding equivalence classes are described by the following proposition:

Proposition 2. *For the equivalence relation defined in Definition 2, there are exactly 8 equivalence classes:*

- *the class 0 consisting of a single tuple* $(0,0,0)$*;*
- *the class R consisting of the tuples* $(x_{\rm R}, 0, 0)$ *with* $x_{\rm R} > 0$*;*
- *the class G consisting of the tuples* $(0, x_{\rm G}, 0)$ *with* $x_{\rm G} > 0$*;*
- *the class B consisting of the tuples* $(0, 0, x_{\rm B})$ *with* $x_{\rm B} > 0$*;*
- *the class RG consisting of the tuples* $(x_{\rm R}, x_{\rm G}, 0)$ *with* $x_{\rm R} > 0$ *and* $x_{\rm G} > 0$*;*
- *the class RB consisting of the tuples* $(x_{\rm R}, 0, x_{\rm B})$ *with* $x_{\rm R} > 0$ *and* $x_{\rm B} > 0$*;*
- *the class GB consisting of the tuples* $(0, x_{\rm G}, x_{\rm B})$ *with* $x_{\rm G} > 0$ *and* $x_{\rm B} > 0$*; and*
- *the class RGB consisting of the tuples* $(x_{\rm R}, x_{\rm G}, x_{\rm B})$ *with* $x_{\rm R} > 0$, $x_{\rm G} > 0$ *and* $x_{\rm B} > 0$.

These are exactly the 8 classes used in the current practically useful implementation of color optical computing (see, e.g., [13, 14, 19, 33, 34]). So, the invariance ideas indeed explain the empirical success of this implementation.

7 Proofs

7.1 Proof of Proposition 1

1°. Let us first prove the result about "and"-operations. It is easy to show that minimum is a scale-invariant "and"-operation. So, to complete the proof, it is sufficient to prove that every scale-invariant "and"-operation is equal to the minimum.

Let $f_\&(a,b)$ be a scale-invariant "and"-operation. Let us consider the case when $0 < a < b < 1$. In general, $f_\&(a,b) \le a$. Let us show that in this case, either $f_\&(a,b) = 0$ or $f_\&(a,b) = a$. We will prove this by contradiction. Indeed, suppose that neither of these two equalities is true and $0 < c \stackrel{\rm def}{=} f_\&(a,b) < a$. Let us consider a monotonic 1-1 function for which $f(0) = 0$, $c < f(c) < a$, $f(b) = b$, and $f(1) = 1$ – we can obtain such function, e.g., by linear interpolation between the point for which the value $f(x)$ is known. For this function $f(a)$, we have $f^{-1}(f_\&(f(a), f(b)) = f^{-1}(f_\&(a,b)) = f^{-1}(c)$, so the invariance condition implies that $f^{-1}(c) = c$, or, equivalently, that $f(c) = c$. However, we know that $f(c) < c$ – a contradiction that shows that $f_\&(a, b)$ cannot take any values strictly in between 0 and a.

So, for every such pair (a, b), we have either $f_\&(a, b) = 0$ or $f_\&(a, b) = \min(a, b)$. Since the function $f_\&(a, b)$ is continuous, it cannot jump from 0 to $\min(a, b) = a > 0$. So, we either have $f_\&(a, b) = 0$ for all such a and b or we have $f_\&(a, b) = \min(a, b)$ for all such pairs.

In the first case of the identically zero function, by taking $a = 1 - 2\varepsilon$ and $b = 1 - \varepsilon$, in the limit $\varepsilon \to 0$, we conclude that $f_\&(1, 1) = 0$, but by definition $f_\&(1, 1) = 1$. Thus, this case is not possible, and $f_\&(a, b) = \min(a, b)$ for all such pairs. In the limit, we get the same formula for all a and b for which $0 \le a \le b \le 1$ – and similar arguments imply that $f_\&(a, b) = \min(a, b)$ when $0 \le b \le a \le 1$. So, every scale-invariant "and"-operation is indeed equal to the minimum.

2°. Let us now prove the result about "or"-operations. It is easy to show that maximum is a scale-invariant "or"-operation. So, to complete the proof, it is sufficient to prove that every scale-invariant "or"-operation is equal to the maximum.

Let $f_\vee(a, b)$ be a scale-invariant "or"-operation. Let us consider the case when $0 < a < b < 1$. In this case, $b \le f_\vee(a, b)$. Let us show that in this case, either $f_\vee(a, b) = b$ or $f_\vee(a, b) = 1$. We will prove this by contradiction. Indeed, suppose that neither of these two equalities is true and $b < c \stackrel{\text{def}}{=} f_\vee(a, b) < 1$. Let us consider a monotonic 1-1 function for which $f(0) = 0$, $f(a) = a$, $f(b) = b$, $c < f(c) < 1$, and $f(1) = 1$ – we can obtain such function, e.g., by linear interpolation between the point for which the value $f(x)$ is known. For this function $f(a)$, we have $f^{-1}(f_\vee(f(a), f(b)) = f^{-1}(f_\vee(a, b)) = f^{-1}(c)$, so the invariance condition implies that $f^{-1}(c) = c$, or, equivalently, that $f(c) = c$. However, we know that $f(c) > c$ – a contradiction that shows that $f_\vee(a, b)$ cannot take any values strictly in between b and 1.

So, for every such pair (a, b), we have either $f_\vee(a, b) = 1$ or $f_\vee(a, b) = \max(a, b)$. Since the function $f_\vee(a, b)$ is continuous, it cannot jump from $\max(a, b) = b < 1$ to 1. So, we either have $f_\vee(a, b) = 1$ for all such a and b or we have $f_\vee(a, b) = \max(a, b)$ for all such pairs.

In the first case of the constant-1 function, by taking $a = \varepsilon$ and $b = 2\varepsilon$, in the limit $\varepsilon \to 0$, we conclude that $f_\vee(0, 0) = 1$, but by definition $f_\vee(0, 0) = 0$. Thus, this case is not possible, and $f_\vee(a, b) = \max(a, b)$ for all such pairs. In the limit, we get the same formula for all a and b for which $0 \le a \le b \le 1$ – and similar arguments imply that $f_\vee(a, b) = \max(a, b)$ when $0 \le b \le a \le 1$. So, every scale-invariant "or"-operation is indeed equal to the maximum.

The proposition is proven.

7.2 Proof of Proposition 2

Clearly, if two tuples $(x'_{\rm R}, x'_{\rm G}, x'_{\rm B})$ and $(x''_{\rm R}, x''_{\rm G}, x''_{\rm B})$ are equivalent, then for each color i, either both values x'_i and x''_i are equal to 0 or both values x'_i and x''_i

are positive. So, to complete the proof, it is sufficient to prove that when for two tuples $(x'_{\rm R}, x'_{\rm G}, x'_{\rm B})$ and $(x''_{\rm R}, x''_{\rm G}, x''_{\rm B})$, for each color i, we have $x'_i = 0$ if and only if $x''_i = 0$, then these tuples are indeed equivalent.

Indeed, for such tuples, we can take $x_i = \max(x'_i, x''_i)$. For colors i for which $x'_i = x''_i = 0$, we can take $d'_i = d''_i = 1$. For colors i for which $x'_i > 0$ and $x''_i > 0$, we can take $d'_i = x'_i/x_i$ and $d''_i = x''_i/x_i$. It is easy to easy that in this cases, we indeed have $x'_i = d'_i \cdot x_i$ and $x''_i = d''_i \cdot x_i$ for all colors i.

The proposition is thus proven.

Acknowledgments

This work was supported by the AT&T Fellowship in Information Technology, by the Institute for Risk and Reliability, Leibniz Universitaet Hannover, Germany, by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) Focus Program SPP 100+ 2388, Grant Nr. 501624329, by the European Union under the project ROBOPROX (No. CZ.02.01.01/00/22 008/0004590), by the Center of Excellence in Econometrics, Faculty of Economics, Chiang Mai University, Thailand, by the Ho Chi Minh City University of Banking, Vietnam, and by Thang Long University, Hanoi, Vietnam.

References

[1] Y. Bekri1, Anatoli Juditsky, and A. Nemirovski, "First order algorithms for computing linear and polyhedral estimates", *Open Journal of Mathematical Optimization*, 2024, Vol. 5, Paper 7.

[2] R. Belohlavek, J. W. Dauben, and G. J. Klir, *Fuzzy Logic and Mathematics: A Historical Perspective*, Oxford University Press, New York, 2017.

[3] G. Carlsson and M. Vejdemo-Johansson, *Topological Data Analysis with Applications*, Cambridge University Press, Cambridge, UK, 2021.

[4] F. L. Chernousko, *State Estimation for Dynamic Systems*, CRC Press, Boca Raton, Florida, 1994.

[5] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, Springer, Cham, Switzerland, 2015.

[6] A. Finkelstein, O. Kosheleva, and V. Kreinovich, "Astrogeometry, error estimation, and other applications of set-valued analysis", *ACM SIGNUM Newsletter*, 1996, Vol. 31, No. 4, pp. 3–25.

[7] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, Cambridge, Massachusetts, 2016.

[8] V. M. Guillemin and S. Sternberg, "An algebraic model of transitive differential geometry", *Bulletin of American Mathematical Society*, 1964, Vol. 70, No. 1, pp. 16–47.

[9] L. Jaulin, M. Kiefer, O. Didrit, and E. Walter, *Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control, and Robotics*, Springer, London, 2012.

[10] P. Joharinad and J. Jost, *Mathematical Principles of Topological and Geometric Data Analysis*, Springer, Cham, Switzerland, 2024.

[11] G. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic*, Prentice Hall, Upper Saddle River, New Jersey, 1995.

[12] D. Knuth, *The Art of Computer Programming. Vol. 2: Seminumerical Algorithms*, Addison-Wesley, Hoboken, New Jersey, USA, 2022.

[13] Yu. P. Kondratenko (ed.), *Research Tendencies and Prospect Domains for AI Development and Implementation*, River Publishers, Aalborg, Denmark, 2024.

[14] O. Kosheleva, V. Kreinovich, V. L. Timchenko, and Yu. P. Kondratenko, "From fuzzy to mobile fuzzy", *Journal of Mobile Multimedia*, 2024, Vol. 20, No. 3, pp. 651–664, DOI: 10.13052/jmm1550-4646.2035.

[15] V. Kreinovich, J. Beck, and H. T. Nguyen, "Ellipsoids and ellipsoid-shaped fuzzy sets as natural multi-variate generalization of intervals and fuzzy numbers: how to elicit them from users, and how to use them in data processing", *Information Sciences*, 2007, Vol. 177, No. 2, pp. 388–407.

[16] B. J. Kubica, *Interval Methods for Solving Nonlinear Constraint Satisfaction, Optimization, and Similar Problems: from Inequalities Systems to Game Solutions*, Springer, Cham, Switzerland, 2019.

[17] S. Li, Y. Ogura, and V. Kreinovich, *Limit Theorems and Applications of Set Valued and Fuzzy Valued Random Variables*, Kluwer Academic Publishers, Dordrecht, 2002.

[18] X. Li, Z. Shao, M. Zhu, and J. Yang, *Fundamentals of Optical Computing Technology: Forward the Next Generation Supercomputer*, Springer, Cham, Switzerland, 2018.

[19] A. Lupo, O. Kosheleva, V. Kreinovich, V. Timchenko, and Yu. Kondratenko, "There is still plenty of room at the bottom: Feynman's vision of quantum computing 65 years later", In: Yu. P. Kondratenko (ed.), *Research Tendencies and Prospect Domains for AI Development and Implementation*, River Publishers, Aalborg, Denmark, 2024, pp. 77–86.

[20] G. Mayer, *Interval Analysis and Automatic Result Verification*, de Gruyter, Berlin, 2017.

[21] J. M. Mendel, *Explainable Uncertain Rule-Based Fuzzy Systems*, Springer, Cham, Switzerland, 2024.

[22] R. E. Moore, R. B. Kearfott, and M. J. Cloud, *Introduction to Interval Analysis*, SIAM, Philadelphia, 2009.

[23] H. T. Nguyen and V. Kreinovich, *Applications of Continuous Mathematics to Computer Science*, Kluwer, Dordrecht, 1997.

[24] H. T. Nguyen, C. L. Walker, and E. A. Walker, *A First Course in Fuzzy Logic*, Chapman and Hall/CRC, Boca Raton, Florida, 2019.

[25] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, Cambridge, U.K., 2013.

[26] V. Novák, I. Perfilieva, and J. Močkoř, *Mathematical Principles of Fuzzy Logic*, Kluwer, Boston, Dordrecht, 1999.

[27] S. Y. Oudot, *Persistence Theory: From Quiver Representations to Data Analysis*, American Mathematical Society, Providence, Rhode Island, USA, 2015.

[28] J. K. Pathos, *Introduction to Topological Quantum Computation*, Cambridge Unioversity Press, Cambridge, UK, 2012.

[29] F. N. Schietzold, J. Urenda, V. Kreinovich, W. Graf, and M. Kaliske, "Why ellipsoids in mechanical analysis of wood structures", *Proceedings of the 9th International Workshop on Reliable Engineering Computing REC'2021*, Taormina, Italy, May 16–20, 2021, pp. 604–614.

[30] K. K. Semenov and V. Kreinovich, "From p-boxes to p-ellipsoids: towards an optimal representation of imprecise probabilities", *Proceedings of the IEEE Symposium on Computational Intelligence for Engineering Solutions CIES'2013*, Singapore, April 16–19, 2013, pp. 149–156.

[31] C. Servin, S. Dick, K. Cohen, and V. Kreinovich (eds.), *Soft Computing, Fuzzy Systems, and Explainable AI*, Springer, Cham, Switzerland, to appear.

[32] I. M. Singer and S. Sternberg, "Infinite groups of Lie and Cartan, Part 1", *Journal d'Analyse Mathematique*, 1965, Vol. XV, pp. 1–113.

[33] V. Timchenko, Yu. Kondratenko, and V. Kreinovich, "Logical platforms for mobile application in decision support systems based on color information processing", *Journal of Mobile Multimedia*, 2024, Vol. 20, No. 3, pp. 679–698, DOI 10.13052/jmm1550-4646.2037

[34] V. Timchenko, V. Kreinovich, and Yu. Kondratenko (eds.), *Optical Color Computing: Fuzzy Logic and Digital Calculations*, River Publishers, Aalborg, Denmark, 2026.

[35] N. Wiener, *Cybernetics, or Control and Communication in the Animal and the Machine*, 3rd edition, MIT Press, Cambridge, Massachusetts, 1962.

[36] L. A. Zadeh, "Fuzzy sets", *Information and Control*, 1965, Vol. 8, pp. 338–353.