

# Human Visual Perception and Kolmogorov Complexity: Revisited

Vladik Kreinovich and Luc Longpré

Department of Computer Science  
University of Texas at El Paso  
El Paso, TX 79968, USA  
emails {vladik,longpre}@cs.utep.edu

## Abstract

Experiments have shown [2] that we can only memorize images up to a certain complexity level, after which, instead of memorizing the image itself, we, sort of, memorize a probability distribution in terms of which this image is “random” (in the intuitive sense of this word), and next time, we reproduce a “random” *sample* from this distribution. This random sample may be *different* from the original image, but since it belongs to the same distribution, it, hopefully, *correctly reproduces the statistical characteristics* of the original image.

The reason why a complex image cannot be accurately memorized is, probably, that our memory is limited. If storing the image itself exhausts this memory, we store its probability distribution instead. With this limitation in mind, we conclude that we cannot store arbitrary probability distributions either, only sufficient simple ones.

In this paper, we show that an arbitrary image is indeed either itself simple, or it can be generated by a simple probability distribution. This result provides a mathematical foundations for the above theoretical explanation of the randomized-memorization phenomena.

**Human visual perception and complexity.** If we see an image, how accurately do we remember it? How long does it take for us to memorize it? Intuitively, the more complicated the image, the longer it takes to memorize it, and the more difficult it is to reproduce it accurately. It is desirable to make this *qualitative* statement *quantitative*, i.e., to somehow gauge this complexity of an image so that we will be able to *predict* the learning time and the reproduction accuracy.

**Kolmogorov complexity and why we need to modify it.** In theoretical computer science, there is a very natural definition of a complexity of a binary string (and in the computer, everything, images included, is represented as a

binary string): the notion of *Kolmogorov complexity* (see, e.g., [1, 3]). Crudely speaking, the Kolmogorov complexity  $C(x)$  of a string  $x$  is the length of the shortest program  $p$  (in some universal language) that generates this string  $x$ .

We cannot, however, use the Kolmogorov complexity itself to predict the time and accuracy of image memorization, because Kolmogorov complexity is not computable [1, 3].

**A modification of Kolmogorov complexity works.** In [2], R. M. Granovskaya, I. Ya. Bereznaya, and A. N. Grigorieva have developed a computable modification of Kolmogorov complexity that predicts the image memorization time reasonably well.

**One experimental fact that needs explanation.** According to the experimental data from [2], when the complexity of an image increases, the time required for its accurate memorization increases as well. However, this relationship only holds while the image is simple (i.e., while its complexity does not exceed a certain constant). As images become more complex, accurate memorization becomes impossible. No matter how long a person looks at the image, he is not able to reproduce it accurately. Moreover, if we repeat this experiment several time, the person will reproduce slightly *different* images. In mathematical-sounding words, it looks like, for such complex images, he grasps some *probability distribution* of which this image is a “random” sample, and then, instead of reproducing the exact image, reproduces “random” samples (images) from this probability distribution.

For example, if we ask a person to memorize a long random walk trajectory, he will reproduce another trajectory that also looks like a random walk, but that will be:

- first, different from the original image, and
- second, different if we repeat this experiment once again.

**How can we explain this fact? From the informal idea of the explanation to a more precise formulation.** Intuitively, the above fact seems to be reasonable to explain: Indeed, the human memory is limited, so when we exceed its abilities, we cannot memorize the image accurately. How can we describe this intuitive idea in more formal terms?

If an image  $x$  is *simple*, i.e., if  $C(x) \leq C_0$  for some constant  $C_0$ , this means that there exists a program  $p$  of length  $l(p) \leq C_0$  for which the compiler  $U$  from the above-mentioned universal language produces  $x$ . Thus, in order to memorize the image, it is sufficient to store this program  $p$ ; then, we can reconstruct the original image as  $x = U(p)$ .

In view of this interpretation, the limitation on  $x$  is probably caused by the fact that only  $C_0$  bits can be stored in the human memory. (This statement is, of course, an oversimplification, because in reality, human memory is not binary, but changing the way of representing numbers does not radically change the Kolmogorov complexity estimates [1, 3].)

So, if  $C(x) > C_0$ , this means that we can no longer memorize  $x$  by storing a program that generates  $x$ . An alternative is to somehow store a *probability distribution*, with the intention to reproduce a random sample from this distribution. How do computers do this nowadays?

All computers and/or programming languages are equipped with standard random number generators that generate a random sequence of independently distributed bits, each of which is equal to 0 and 1 with equal probability  $1/2$ . More complicated random number generators (RNG), which simulate different probability distributions, are designed by applying some algorithms to the result of this standard RNG.

With this in mind, a natural way to memorize a complex image is to memorize the program  $f$  that, being applied to a certain “random” sequence  $s$ , generates the original sequence  $x$ :  $f(s) = x$ . In this case, when we reconstruct the image, we run the standard RNG, generate a sequence  $s'$ , and apply the function  $f$  to this sequence. The resulting image  $x' = f(s')$  may be different from the original image  $x$ , but it sure belongs to the same probability distribution as the original one.

**A crucial question.** According to the above explanation, in order to store a complex image  $x$ , we must store the program  $f$  that transforms some random sequence  $s$  into  $x$ . When storing  $f$ , we encounter the same storage problem: that the length of  $f$  has to be limited.

So, for the above explanation to work, we must be sure that for every string  $x$ , either this string itself is simple, or it can be obtained from a random string by some simple program.

In this paper, we will show that, under an appropriate formalization, this is indeed true.

**Towards formalization.** In order to formalize the above dichotomy, we must describe what “ $s$  is random” means. This formalization is one of the main reasons why R. Solomonoff, G. Chaitin, and A.N. Kolmogorov introduced the notion of what is now called Kolmogorov complexity in the first place, and the first (and one of the most natural) formalization of this notion is the notion of *C-randomness*: a sequence  $x$  is  $C$ -random if  $C(x) \geq l(x) - C$ , where  $C$  is an arbitrary positive integer.

With this definition, we arrive at the following formalization:

**Definition.** Let  $C_0$  be a positive integer, let  $U$  be a universal program, and let  $C(x) = C_U(x)$  denote Kolmogorov complexity with respect to  $U$ .

- A binary sequence  $x$  is called  $C_0$ -simple if  $C(x) \leq C_0$ .
- A binary sequence  $s$  is called  $C_0$ -random if  $C(s) \geq l(s) - C_0$ .
- We say that a sequence  $x$  is obtained from  $s$  by a  $C_0$ -simple program if  $x = U(f|s)$  for some program  $f$  of length  $l(f) \leq C_0$ .

**Proposition.** *There exists an integer  $C_0$  for which every binary sequence  $x$  is:*

- *either  $C_0$ -simple,*
- *or can be obtained from some  $C_0$ -random sequence  $s$  with  $l(s) \leq l(x) + C_0$  by applying a  $C_0$ -simple program.*

*Comment.* On the *fundamental* level, this result provides a mathematical foundations for the above theoretical explanation of the randomized-memorization phenomena. explains the above experimental fact. To make this result more *practical*, we must use a computable modification of Kolmogorov complexity instead of the Kolmogorov complexity itself.

**Proof.** Our first comment is that out of the two alternatives provided by the theorem, the first is a particular case of the second one. Indeed, the fact that the sequence  $x$  is  $C_0$ -simple means that  $x$  can be generated by a program  $p$  of length  $l(p) \leq C_0$ . We can view it as a degenerate case of the second alternative, in which the random sequence  $s$  is of length 0.

In view of this comment, it is sufficient to prove that the statement expressed by the second alternative always holds, i.e., that for every string  $x$ , there exists a  $C_0$ -random sequence  $s$  of length  $l(s) \leq l(x) + C_0$  and a  $C_0$ -simple program  $f$  for which  $f(s) = U(f|s) = x$ .

Indeed, let  $x$  be an arbitrary string, and let  $C(x)$  be its Kolmogorov complexity. By definition of the Kolmogorov complexity, there exists a program  $p$  for which  $x = U(p)$  and  $l(p) = C(x)$ .

If we start with  $p$  instead of  $x$ , we can similarly conclude that there exists a program  $q$  for which  $p = U(q)$  and  $l(q) = C(p)$ . Hence,  $x = U(p) = U(U(q))$ . If we use the notation  $\circ$  for composition, we can conclude that  $x = U \circ U(q)$ . From this equality, we can conclude that  $l(q) \geq C_{U \circ U}(x)$ .

Since  $U$  is a universal program, there exists a constant  $C_1$  for which, for every  $x$ ,  $C_{U \circ U}(x) \geq C_U(x) - C_1$ . Therefore,  $l(q) \geq C_{U \circ U}(x) \geq C_U(x) - C_1$ . But, by definition of  $p$ ,  $C_U(x) = l(p)$ , and by definition of  $q$ ,  $l(q) = C(p)$ . Therefore, the above inequality turns into  $C(p) \geq l(p) - C_1$ , i.e., it proves that  $p$  is  $C_1$ -random. So, we can take this  $p$  as  $s$ .

It is known (and easily provable) that there exists a constant  $C_2$  for which  $C(x) \leq l(x) + C_2$  for every sequence  $x$  (e.g, we can always generate  $x$  by typing something like `writeln(x)`, so, we always have a program  $p_0$  of size  $l(x) + C_2$  to generate  $x$ , and thus, the length  $C(x)$  of the shortest such program cannot be larger than  $l(x) + C_2$ ). Hence,  $l(p) \leq l(x) + C_2$ . So, if we take  $C_0 = \max(C_1, C_2)$ , we can conclude that  $l(s) \leq l(x) + C_0$  and that  $s$  is  $C_0$ -random.

Now,  $x = U(s)$ , i.e., a simple  $f (= U)$  indeed suffices to compute  $x$  from  $s$ . The proposition is proven.

**Acknowledgments.** This work was supported in part by NASA under cooperative agreement NCCW-0089, by NSF under grants No. DUE-9750858 and EEC-9322370, and by the Future Aerospace Science and Technology Program

(FAST) Center for Structural Integrity of Aerospace Systems, effort sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant number F49620-95-1-0518.

The authors are thankful to Dima Grigoriev and Alla Grigorieva for useful references and explanations.

## References

- [1] C. Calude, *Information and randomness: An algorithmic perspective*, Springer-Verlag, Berlin, 1994.
- [2] R. M. Granovskaya, I. Ya. Bereznaya, and A. N. Grigorieva, *Perception of form and forms of perception*, Erlbaum, Hillsdale, NJ, 1987.
- [3] M. Li and P. Vitányi, *An introduction to Kolmogorov complexity and its applications*, Springer-Verlag, N.Y., 1997.