EXCERPTED FROM

STEPHEN WOLFRAM

A NEW KIND OF SCIENCE

SECTION 10.6

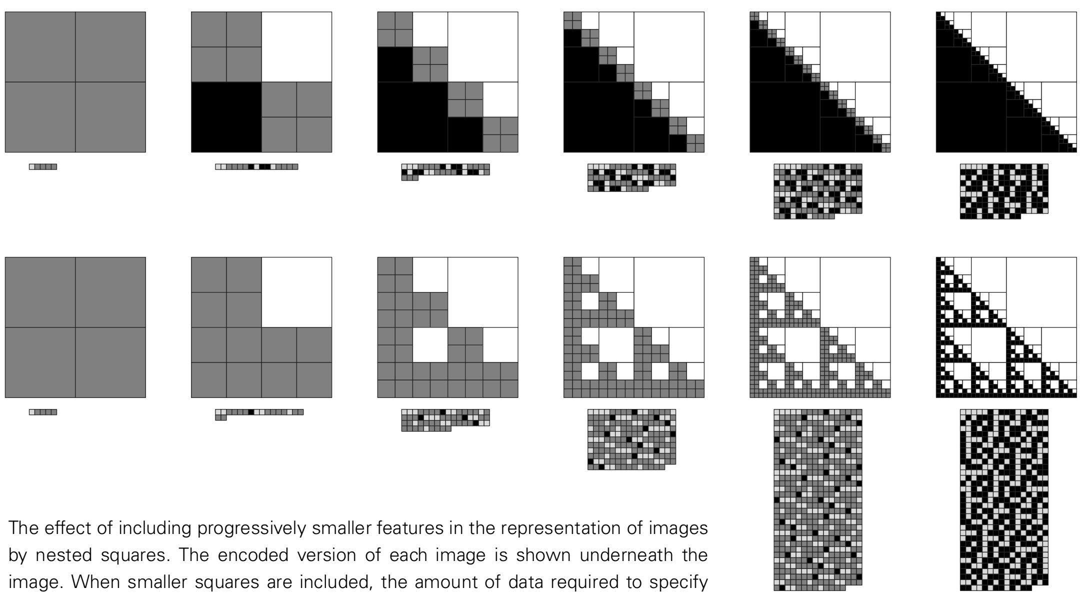
Irreversible Data Compression

as we have discussed them in this section come even close to finding such short descriptions. And as a result, at least with respect to any of these methods all we can reasonably say is that the behavior we see seems for practical purposes random.

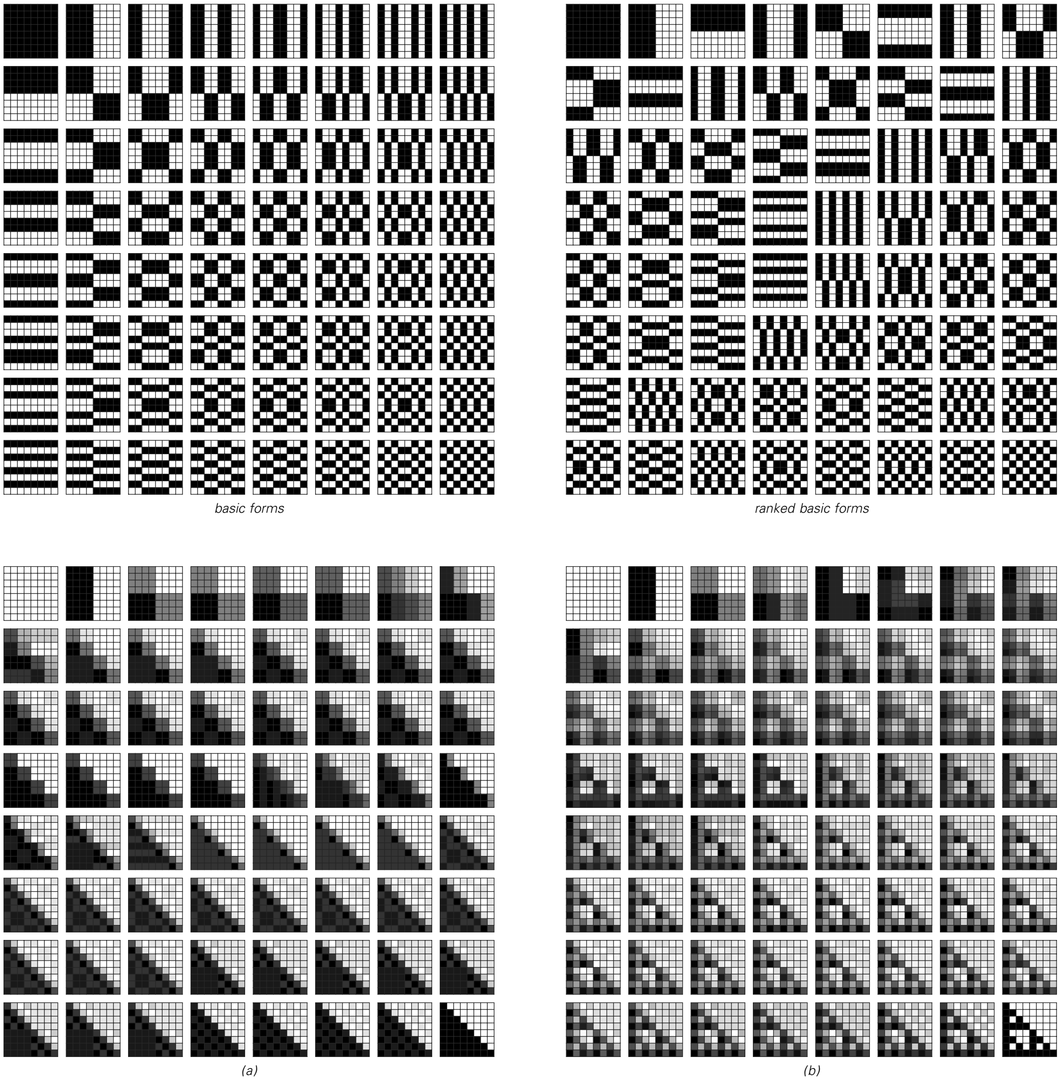
Irreversible Data Compression

All the methods of data compression that we discussed in the previous section are set up to be reversible, in the sense that from the encoded version of any piece of data it is always possible to recover every detail of the original. And if one is dealing with data that corresponds to text or programs such reversibility is typically essential. But with images or sounds it is typically no longer so necessary: for in such cases all that in the end usually matters is that one be able to recover something that looks or sounds right. And by being able to drop details that have little or no perceptible effect one can often achieve much higher levels of compression.

In the case of images a simple approach is just to ignore features that are smaller than some minimum size. The pictures below show



The effect of including progressively smaller features in the representation of images by nested squares. The encoded version of each image is shown underneath the image. When smaller squares are included, the amount of data required to specify the image increases rapidly.



Examples of how images can be built up by adding together basic forms consisting of so-called two-dimensional Walsh functions. On the top left the basic forms are given in so-called sequency order. On the top right they are reordered roughly so as to go systematically from coarser to finer. In the bottom arrays of pictures each successive picture is obtained by adding in the corresponding basic form with an appropriate weight. The basic forms shown here have the property of being orthogonal, so that the weight for each form can be deduced simply by multiplying the original image by that form. Note that the forms involve numerical values -1 and +1, corresponding to cells colored white and black. The images shown here are all rescaled so that smallest values are white and largest black. The JPEG method of image compression uses an approach similar to the one shown here, though with basic forms that have continuous levels of gray, rather than just black and white.

what happens if one divides an image into a collection of nested squares, but imposes a lower limit on the size of these squares. And what one sees is that as the lower limit is increased, the amount of compression increases rapidly—though at the cost of a correspondingly rapid decrease in the quality of the image.

So can one do better at maintaining the quality of the image? Various schemes are used in practice, and almost all of them are based on the idea from traditional mathematics that by viewing data in terms of numbers it becomes possible to decompose the data into sums of fixed basic forms—some of which can be dropped in order to achieve compression.

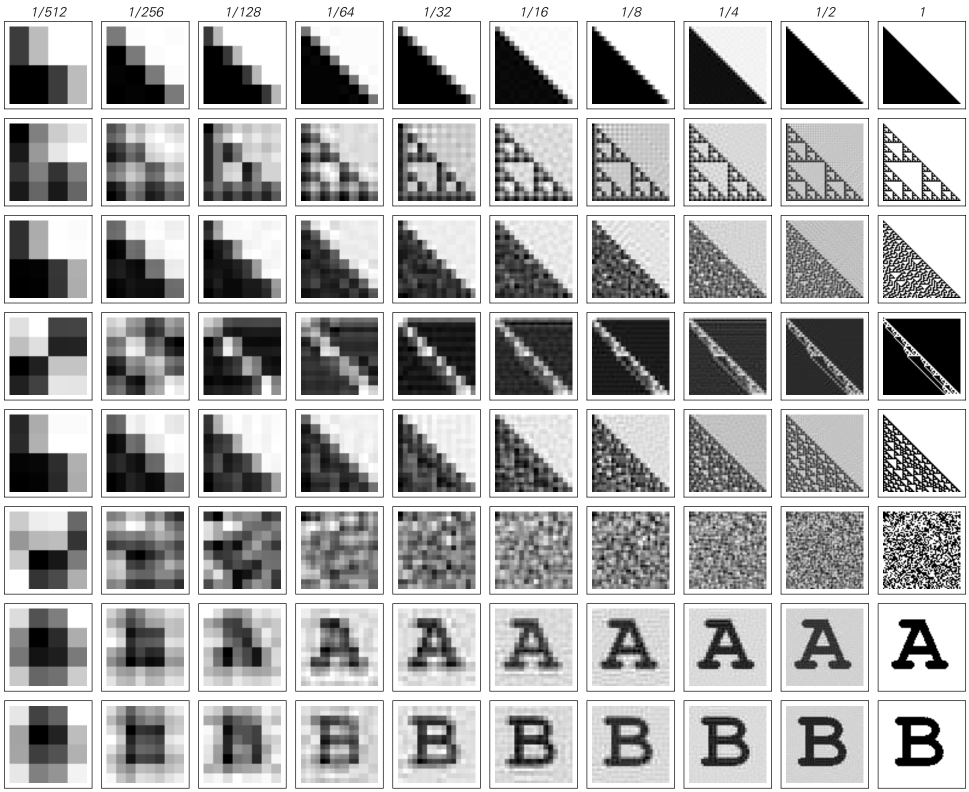
The pictures on the previous page show an example of how this works. On the top left is a set of basic forms which have the property that any two-dimensional image can be built up simply by adding together these forms with appropriate weights. On the top right these forms are then ranked roughly from coarsest to finest. And given this ranking, the arrays of pictures at the bottom show how two different images can be built up by progressively adding in more and more of the basic forms.

If all the basic forms are included, then the original image is faithfully reproduced. But if one drops some of the later forms—thereby reducing the number of weights that have to be specified—one gets only an approximation to the image. The facing page shows what happens to a variety of images when different fractions of the forms are kept.

Images that are sufficiently simple can already be recognized even when only a very small fraction of the forms are included—corresponding to a very high level of compression. But most other images typically require more forms to be included—and thus do not allow such high levels of compression.

Indeed the situation is very much what one would expect from the definition of complexity that I gave two sections ago. The relevant features of both simple and completely random images can readily be recognized even at quite high levels of compression. But images that one would normally consider complex tend to have features that cannot be recognized except at significantly lower levels of compression.

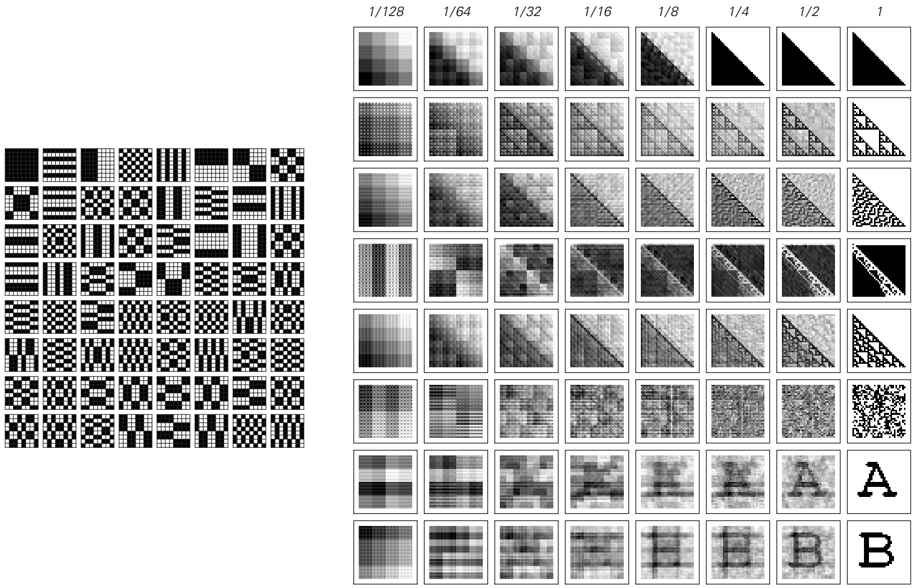
All the pictures on the facing page, however, were generated from the specific ordering of basic forms shown on the previous page. And

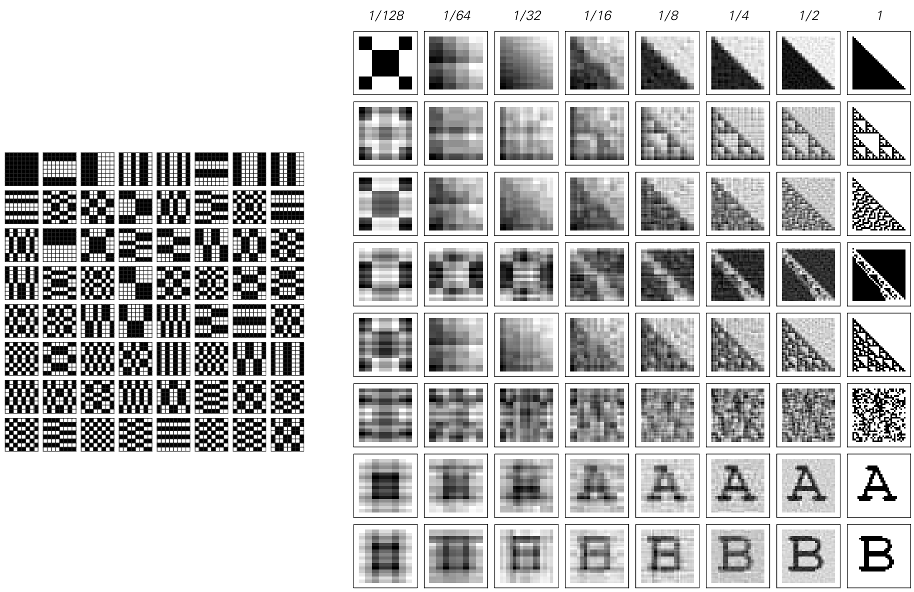


Examples of images obtained by keeping only certain fractions of the complete set of basic forms. In the case of both simple and completely random images, many features are recognizable even with fairly few basic forms—implying that a highly compressed representation can be given.

one might wonder whether perhaps some other ordering would make it easier to compress more complex images.

One simple approach is just to assemble a large collection of images typical of the ones that one wants to compress, and then to order the basic forms so that those which on average occur with larger weights in this collection appear first. The pictures on the next page show what happens if one does this first with images of cellular automata and then with images of letters. And indeed slightly higher levels of compression are achieved. But whatever ordering is used the fact seems to remain that images that we would normally consider complex still cannot systematically be compressed more than a small amount.





Results obtained by deducing optimal orderings of basic forms from collections of images of cellular automata (top) and letters (bottom). The orderings of basic forms are shown on the left, in each case starting with those whose weights are largest in absolute value when averaged over the collection of images. Note that the orderings are shown for 8 × 8 basic forms, while the actual images are 32 × 32. The orderings are deduced respectively from images of the 256 elementary cellular automata, and the 52 upper and lower case letters.