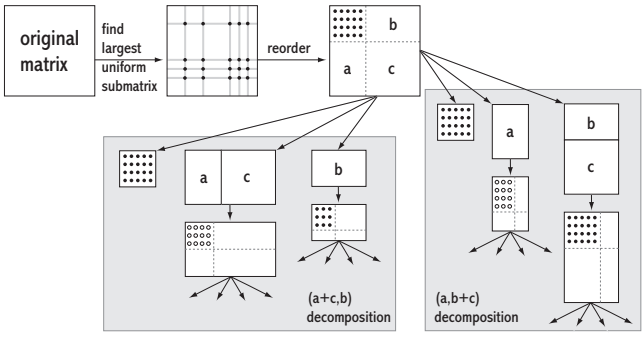
A compression-boosting transform for 2D data

Qiaofeng Yang Stefano Lonardi

Department of Computer Science and Engineering
University of California
Riverside, CA 92521

Every day massive quantities of two-dimensional data are produced, stored and transmitted. Matrices over finite alphabets are used to represent all sorts of information, like digital images, graphs, database tables, geometric objects, etc. From the compression standpoint, two-dimensional data has to be treated differently than one-dimensional data. In order to obtain good compression, one has to exploit the dependencies (or equivalently, expose the redundancies) both between the rows and the columns of the matrix.

In this paper, we present an invertible transform for two-dimensional data which has the objective of reordering the matrix so it will improve its (lossless) compression at later stages. Given a binary matrix to be transformed, the transform involves first searching for the largest *uniform submatrix*, that is, a submatrix solely composed by the same symbol (either 0 or 1) induced by a subset of rows and columns (which are not necessarily contiguous). Then, the rows and the columns are reordered such that the uniform submatrix is moved to the left-upper corner of the matrix. The transform is recursively applied on the rest of the matrix. One step of the forward transform is illustrated in the figure. The recursion is stopped when the partition produces a matrix which is smaller than a predetermined threshold.



One step of the forward transform.

The computational cost of the forward transform depends on the complexity of finding the largest uniform submatrix. Although the latter problem turns out to be **NP**-hard, we present a relatively simple randomized algorithm that has good performance in practice. The inverse transform (decompression) is fast and can be implemented in linear time in the size of the matrix.

The effects of the transform on the compressibility of two-dimensional data is studied empirically by comparing the performance of `gzip` and `bzip2` before and after the application of the transform on several inputs. The preliminary results show that the transform boosts compression.

Proceedings of the 2005 Data Compression Conference (DCC'05)
1068-0314/05 $ 20.00 **IEEE**