Controlled Data Compression with Adaptive hp-FEM

PAVEL SOLIN, DAVID ANDRS
Department of Mathematical Sciences, The University of Texas at El Paso (UTEP)
http://hpfem.math.utep.edu/

PDEs, FEM, and hp-FEM

Many important natural processes, such as the weather, flow of liquids, deformation of solid bodies, heat transfer, effects of electromagnetic fields, and others, are described by *partial differential equations (PDEs)*. PDEs are too hard to be solved exactly, and therefore they have to be solved approximately by means of sophisticated numerical (computational) methods. The *Finite element method (FEM)* is the most widely used numerical method for the approximate solution of PDEs. It is based on partitioning the computational domain into small and geometrically simple objects, *finite elements*, where the solution to the PDE (such as fluid velocity, density, pressure, displacement, stress, temperature, electromagnetic field, etc.) is approximated by polynomials. The piecewise-polynomial approximation is characterized by a finite number of unknown real coefficients. These constants are called *degrees of freedom (DOF)*. In the hp-FEM (also called hp-version of the FEM), both the size h and polynomial degree p of elements moreover are adapted automatically in order to maximize the convergence speed. By maximizing convergence speed we mean to reduce the approximation error as fast as possible with respect to the number of DOF used. It has been proven that the hp-FEM can attain *exponential rate of convergence*. This is the fastest known convergence rate where the error ERR of the approximation decays extremely fast, obeying the relation $ERR \approx e^{-DOF}$.

Application to Image Compression

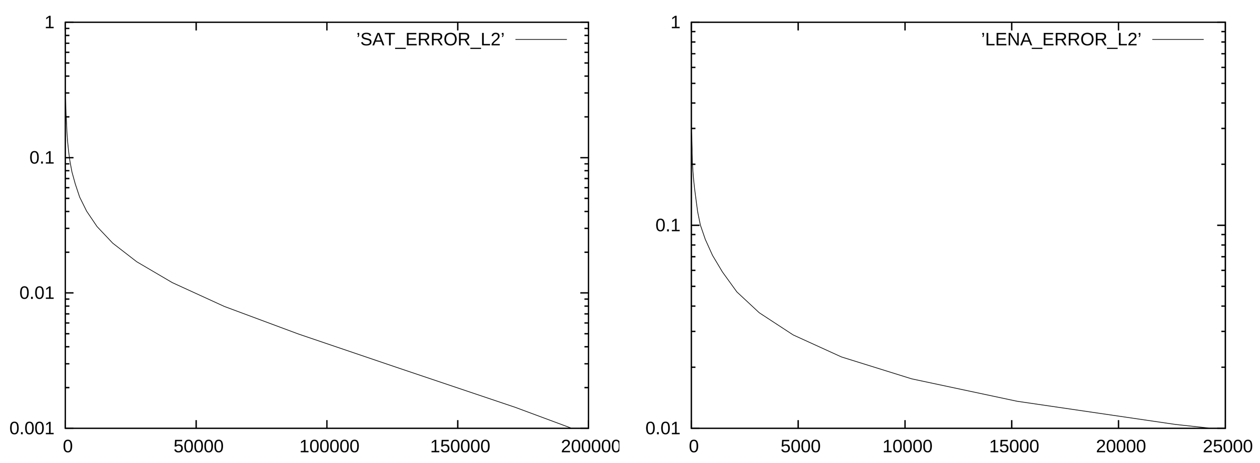
Both the FEM and hp-FEM can be applied to both 2D and 3D image compression naturally: The computational domain Ω is the rectangle (in 2D) or hexahedron (in 3D) containing the image. A greyscale image is a piecewise-constant function defined in Ω, with values between 0 and 255 in pixels. A color image consists of three such functions for the red, green, and blue components. The image plays the role of an *exact solution* to a PDE (which never is known when a PDE is solved, but we are not solving any PDE this time). It follows from here that our algorithm is capable of compressing continuous data as well, not only data which are constant in pixels. The adaptive algorithm starts with an extremely coarse mesh, typically just one single element. One step of the algorithm looks as follows: (1) determine elements with largest projection error (2) determine optimal refinement of every such element (3) refine the elements in question (4) project the image on the new mesh.

Comparison to Existing Techniques

Our algorithm does not contain any heuristics which are typical for image compression algorithms such as JPEG, it can compress more general data than piecewise-constant functions (for example, continuous functions), and it controls the error in the compression (i.e., compression data loss). Based on the prescribed error tolerance, the algorithm can be both lossy and lossless. The compressed images are stored as continuous, piecewise-polynomial functions, and thus existing visualization methods and software can be applied *directly to the compressed images*. This is impossible with standard image compression techniques (JPEG, GIF, and others). On the other hand, our algorithm is still slower than standard image compression algorithms, and we need to optimize it.

Error Control and Convergence

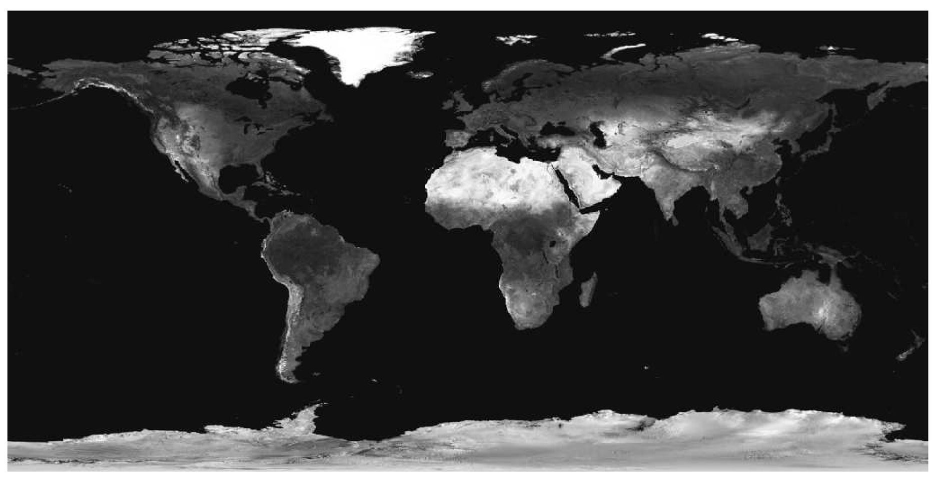
The following graphs show the convergence for both examples.



Convergence of the adaptive process for the satellite photo (left) and Lena (right). Horizontal and vertical axes represent DOF and ERR, resp.

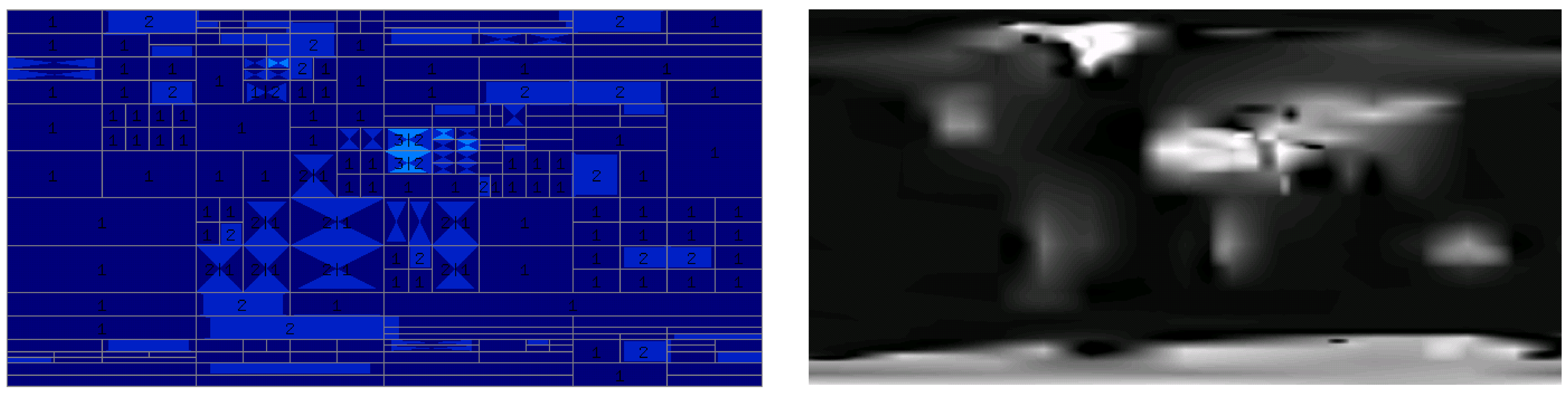
Example: Compression of Satellite Image of Earth

Performance of the adaptive algorithm can be illustrated on a high-resolution satellite image of Earth (courtesy NASA).

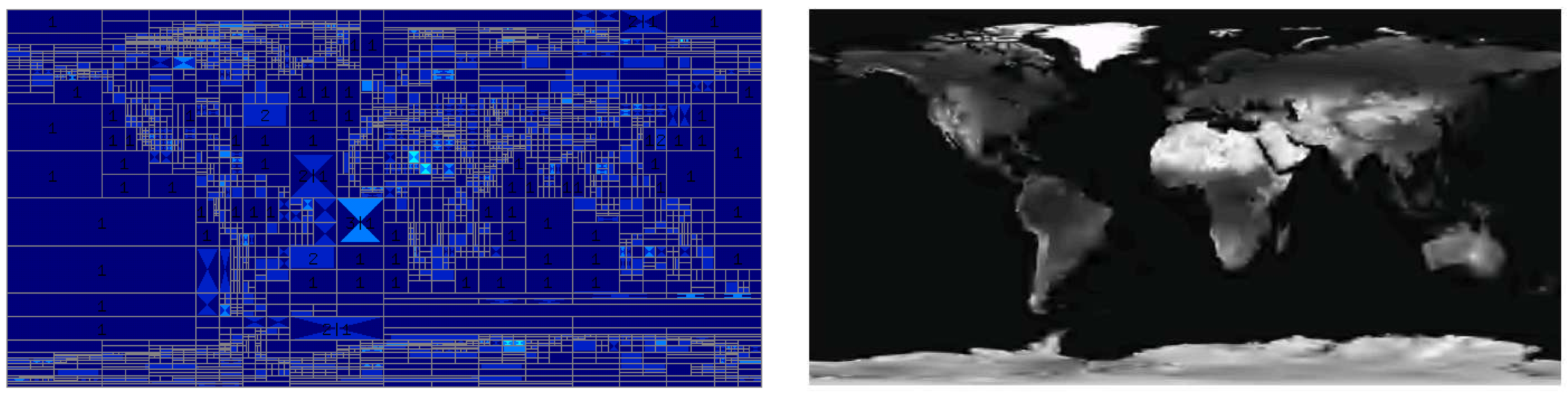


Satellite photograph of Earth, size 1024×512 pixels.

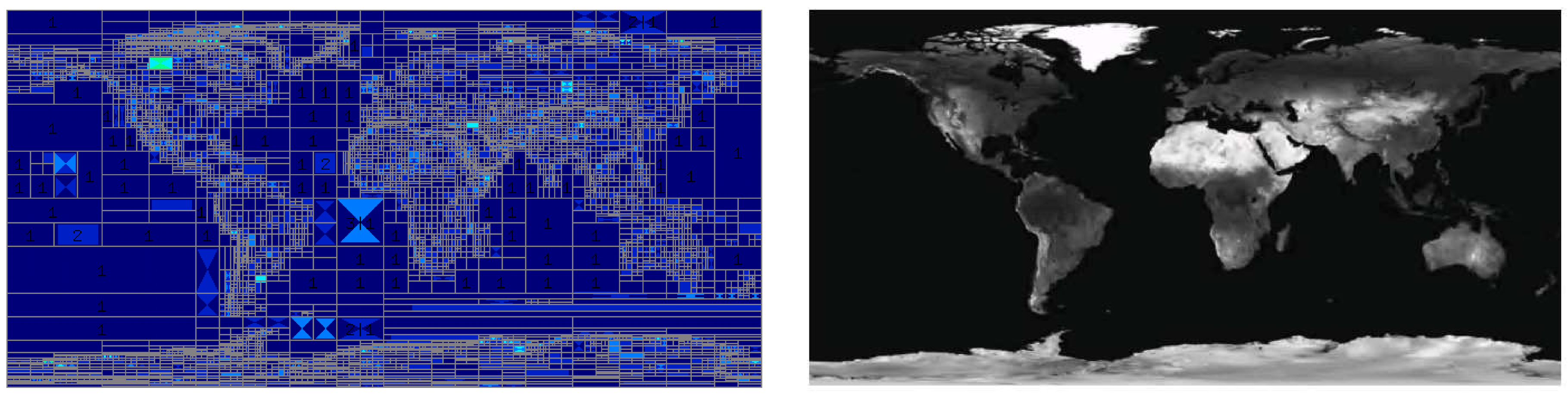
Let us look at approximations corresponding to relative error ERR of 18%, 6%, 3%, and 1% in the L^2-norm below. The symbol “DOF” stands for *degrees of freedom* (parameters defining the approximation). Notice that our algorithm leaves huge elements in areas where ”nothing happens” while small elements only are used to resolve small-scale phenomena. This is highly economical. In contrast to that, JPEG always subdivides the image into small cella of 8×8 pixels.



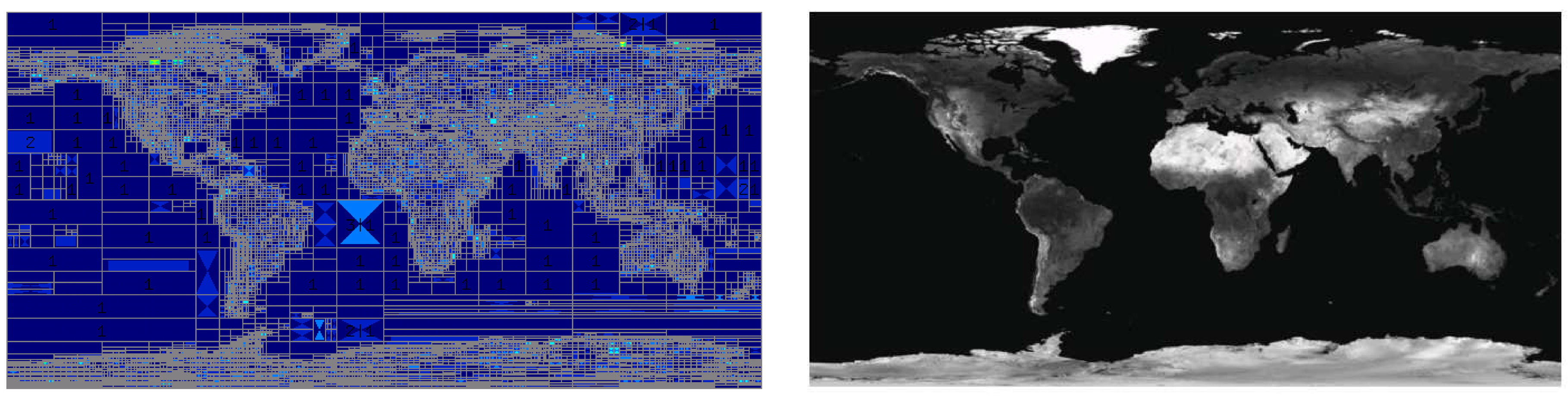
Finite element mesh and approximation, $DOF = 419$, $ERR = 18.2095\%$.



Finite element mesh and approximation, $DOF = 3778$, $ERR = 6.3895\%$.



Finite element mesh and approximation, $DOF = 12125$, $ERR = 3.0938\%$.



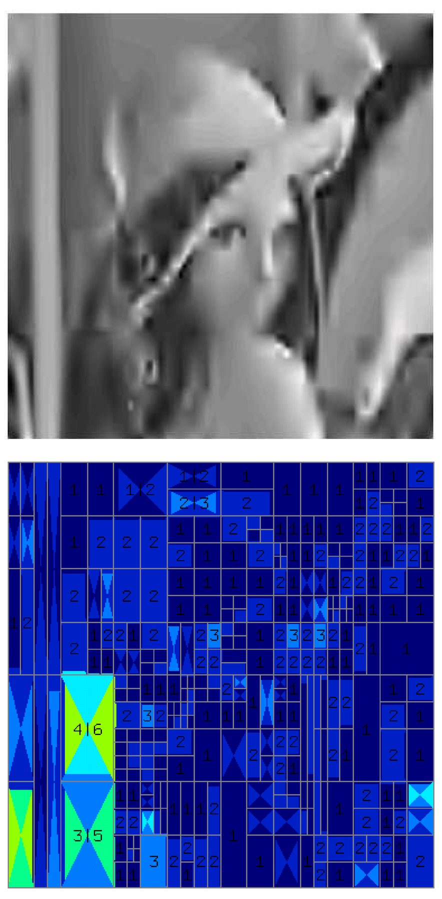
Finite element mesh and approximation, $DOF = 40901$, $ERR = 1.1916\%$.

Example: Lena

This is a traditional benchmark problem in the image compression community, (upper part of) photo of a November 1972 Playmate Lena Sjööblom.

Lena, size 512×512 pixels.

This image contains multiple phenomena that cause problems to image compression algorithms.



$DOF = 420$, $ERR = 10.0344\%$.

$DOF = 2129$, $ERR = 4.7012\%$.

$DOF = 22372$, $ERR = 1.0449\%$.

References

[1] P. Solin, *Partial Differential Equations and the Finite Element Method*, 504 pages, J. Wiley & Sons, ISBN 0-471-72070-4, 2005.

[2] P. Solin, K. Segeth, I. Dolezel: *Higher-Order Finite Element Methods*, 408 pages, Chapman & Hall/CRC Press, July 2003, ISBN 158488438X.

[3] P. Solin, J. Cerveny, I. Dolezel: Arbitrary-Level Hanging Nodes and Automatic Adaptivity in the hp-FEM, MATCOM, published online, doi:10.1016/j.matcom.2007.02.011.

[4] P. Solin, T. Vejchodsky: Higher-Order Finite Elements Based on Generalized Eigenfunctions of the Laplacian, Int. J. Numer. Methods Engrg, published online, doi:10.1002/nme.2129.