

# Quality and Resource Controlled Transmission of Images

Uwe Rauschenbach  
Randolf Schultz  
Heidrun Schumann

University of Rostock  
Computer Science Department  
D-18051 Rostock  
Germany

Email: {urausche, rschultz, schumann}@informatik.uni-rostock.de

**Abstract.** This paper presents an image transmission protocol which allows to transmit images controlled by resources and quality demands. Depending on these parameters, the reduction of image data, the selection and the configuration of a compression algorithm are done automatically. The automatic selection mechanism is based on the results of an investigation of image compression algorithms we conducted in order to find out the strengths, weaknesses and configurability of JPEG, fractal and wavelet-based compression.

**Keywords:** Image transmission, Image compression, Image quality, Protocol

## 1 Introduction

Mobile computers, wireless data transmission and globally distributed information spaces (e.g., the WWW) are widely available now and create new technological challenges. Especially the transmission of image data is very bandwidth-demanding, and techniques of image data reduction and compression have to be used. This paper deals with the transmission of images from a server to a mobile client computer, where one has to cope with lots of constraints limiting resources, like bandwidth and processing power. The paper addresses two aspects of the problem:

First, we discuss the results of an investigation of image compression algorithms we conducted in order to find out the strengths and weaknesses of several lossy compressors and their configurability.

Second, we propose an image transmission protocol which allows to transmit images controlled by resources and quality demands. Depending on these parameters, the reduction of image data, the selection and the configuration of a compression algorithm are done automatically.

## 2 Investigation of Image Compression Algorithms

### 2.1 Motivation

Our goal was to develop a control mechanism that accepts quality and resource parameters and calculates a set of configuration parameters for a certain compression algorithm so that the resulting image quality and the use of resources matches the given values. This mechanism is intended to be the core of a protocol for the quality and resource controlled transmission of images.

There are a lot of different compression algorithms for images. Some are standardized, accepted and widely used like JPEG. Other new approaches, namely fractal compression and compression using wavelets, are relatively unknown regarding their strengths and weaknesses. To select a major compression algorithm to use for the transmission of images over a low-bandwidth network, we investigated and compared JPEG, wavelet and fractal based compression.

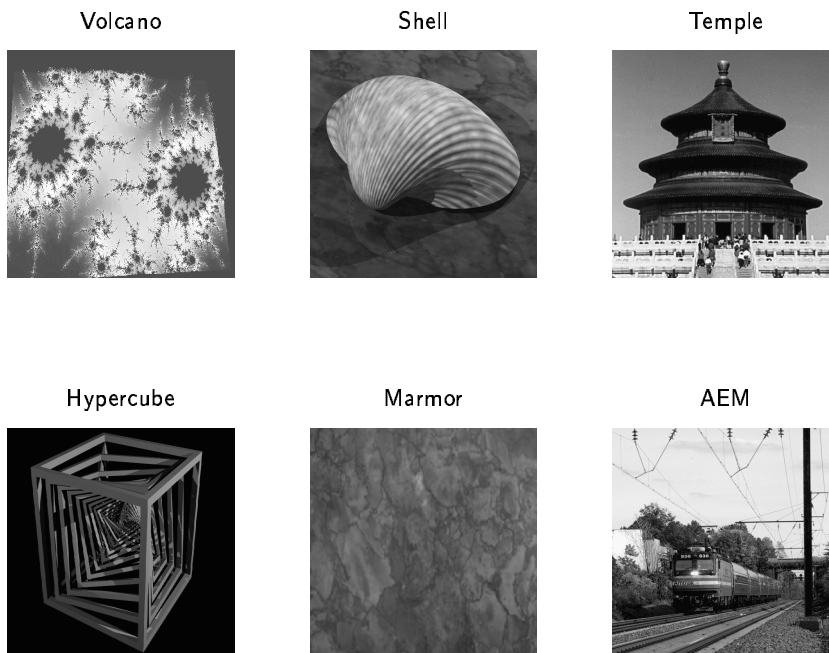
Lossy algorithms achieve the best compression ratio. Certain parameters control the compression ratio and thereby the loss of quality caused by the compression/decompression process. However, they do not directly correspond to the image quality or to the amount of resources saved by the compression. Thus, our experiments had to clarify this correlation.

### 2.2 Evaluation Criteria and Procedure

Regarding image transmission over low bandwidth links, we used the following criteria to evaluate a compression algorithm's quality:

- maximum achievable compression ratio
- compression and decompression costs (CPU time)
- subjective image quality at a certain compression ratio
- dependence on image content
- configurability
- support of progressive or hierarchical encoding
- symmetry of costs for compression and decompression

It is difficult to judge the quality of images observed by humans. Often, "technical" measures like the signal-to-noise ratio or comparisons of the statistical distribution of colors are used. The drawback of these signal characteristics is that they don't take the special capabilities of the human perceptual system into consideration. PQS [1], a quality judgement system which does consider these, is unfortunately limited to monochrome images. Thus, we used our own



**Figure 1:** The test images

subjective criteria to judge image quality:

- good (no visible alterations)
- medium (visible artifacts)
- poor (heavy distortions)

We selected a set of six images (see figure 1) with different characteristics as the test data set. Attention has been paid to cover a wide range of features, e.g. scanned photographs (AEM, Temple), computer generated images (Hypercube, Shell), images with fine lines (AEM), Fractals (Volcano) and textures (Marmor).

We used the JPEG implementation of the Independent JPEG Group, the wavelet implementation EPIC, which is based on Daubechie's wavelet, and the example implementation of a fractal compressor by Yval Fisher. Each compressor can be configured by one or more parameters, which influence quality and compression ratio. In the case of more than one parameter, the parameter which had the greatest influence on the compression ratio has been selected in preliminary tests. A test suite has been implemented which iterates the selected parameter for each

compressor and each image within a predefined range and with a predefined step size. CPU time and compression ratio for each compressor/decompressor run has been recorded, and inverted difference images between the original image and the particular compressed/decompressed images have been computed for quality judgement by a human observer.

### 2.3 Evaluation results

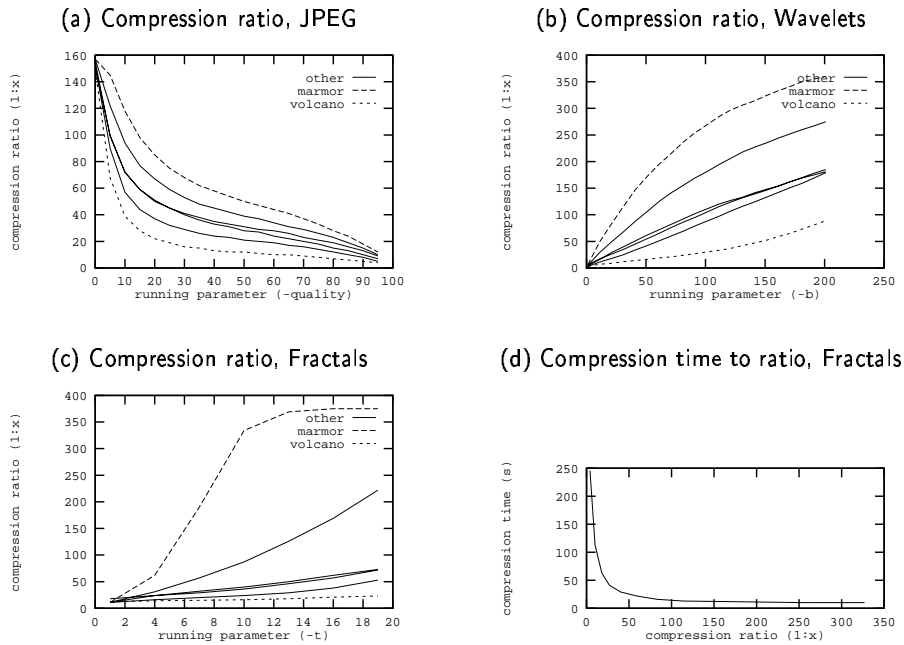
criteria	JPEG	Wavelet	Fractal
maximum compression ratio	1:160	1:350	1:375
compression ratio, quality: good	up to 1:30	up to 1:30	up to 1:25
compression ratio, quality: medium	up to 1:70	up to 1:90	up to 1:50
compression ratio, quality: bad	up to 1:160	up to 1:350	up to 1:375
compression time (factor)	1	*5	*20 (average)
decompression time (factor)	1	*3	*5
dependence on image	medium	big	big
configurable	yes	yes	yes
detail on demand, progressive	yes	yes	no
symmetric	yes	no	no

**Table 1:** Results of the evaluation

Table 1 shows the results of the evaluation:

- JPEG is the fastest and most stable compression method (i.e. independent from image data to a high degree, see figure 2a).
- JPEG is the only symmetric method.
- Wavelets and fractal compression are several times slower than JPEG during compression as well as during decompression.
- Wavelets and fractal compression are much more image-dependent than JPEG
  - compression ratio breaks down on fine structures (c.f. performance measures for the "volcano" test image, figures 2b and 2c).
  - better compression than JPEG on textures (c.f. performance measures for the "marmor" test image, figures 2b and 2c).
- The compression time of a fractal encoder depends to a high degree on the compression ratio (see figure 2d).

- Wavelets deliver better image quality at high compression ratios than JPEG and fractal compression - however, at the cost of a long computing time.



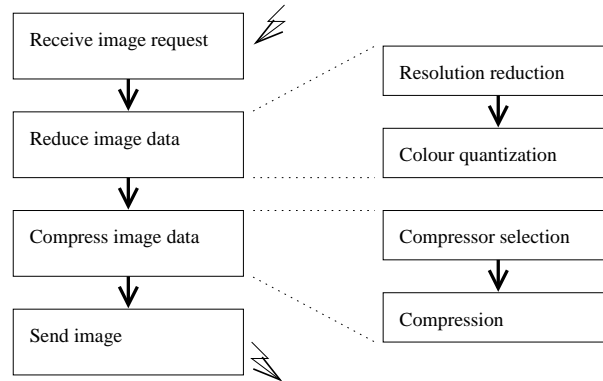
**Figure 2:** Experiment results as diagrams

Tests done subsequently using other implementations than the three mentioned above confirm the trends we found. Since JPEG is the fastest algorithm and allows better prediction of the compression ratio given the image quality (and vice versa) than wavelet and fractal based compression, we selected this compression method for use in our image transmission protocol, which we will describe in the rest of this paper.

### 3 Image transmission protocol

#### 3.1 Motivation

In a mobile environment, transmission bandwidth, client display resources and client processing power are limited. Transmission bandwidth can be saved by using image compression methods, and the images must be adapted to the client's



**Figure 3:** Tasks of the image server

display resources. Our quality and resource controlled image transmission protocol has been designed to consider these aspects.

The basic idea is to do all necessary reduction at the server side in order to save transmission bandwidth and client processing power. In a first step, all those parts of the image can be removed which can't be displayed at the client. This includes the reduction of the resolution as well as the reduction of the number of colors. In a second step, lossy or lossless data compression can be applied in order to decrease transmission bandwidth demands further. When applying lossy data compression, quality requirements of the user should be accepted and mapped to the parameter values of the compression algorithms. Our protocol is designed to adapt the image to the client's display before compressing it, and to automatically select and configure an appropriate compression algorithm.

### 3.2 Features

For an effective image transmission, the protocol must have the following basic features:

- To reduce communication costs, the protocol must use compression techniques which can be either selected automatically depending on the image type (e.g., JPEG for photographs or JBIG for scanned documents) or explicitly by a client request.
- Images must be cropped, scaled, resampled and quantized at the server side according to the client's display resources in order to save communication bandwidth and to reduce the workload on the client system when displaying the image, as stated above.

- It must be easy to extend the knowledge base of the protocol by new compression techniques.
- The protocol should support the separate transmission of several resolution or quality levels as they are offered by some compressors (e.g., Progressive JPEG). This allows the use of detail-on-demand and progressive refinement techniques, which lead to shorter response times and enable a high degree of user control over the transmission process.

### 3.3 Description of the Protocol

**Request processing and structure.** The basic tasks of the image server are illustrated in figure 3. As the protocol works with knowledge about the display parameters of the client in order to provide a default mechanism, we must distinguish two types of requests: contact request and image request. When the client contacts the server for the first time, it sends the *contact* request which contains the following information:

- Client address
- List of available decompressors
- Screen size and resolution
- Colour depth
- Palette

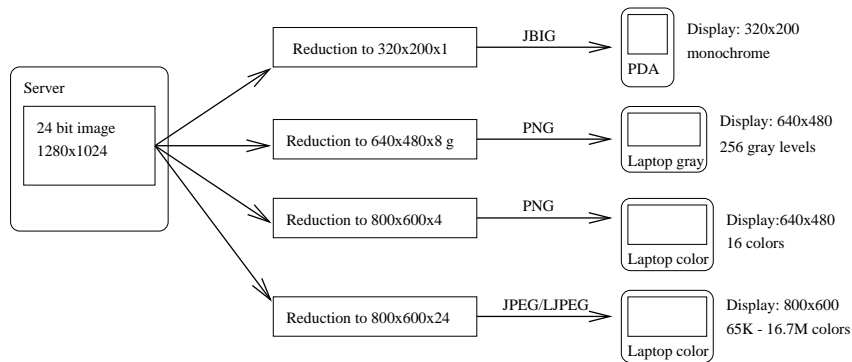
To fetch an image, the client sends an *image request* which is structured as follows:

- Image identifier
- Client address
- Requested quality or compression ratio
- Flag: successive refinement requested
- Requested compressor (optional)
- Requested image resolution (optional)
- Requested image size (optional)
- Requested cropping coordinates (optional)
- Requested color depth (optional)

- Requested palette (optional)

Where the first four fields of the request are mandatory, the optional fields can be used by the client to override default values and the automatic selection process. This facility offers the client a flexible way to control the image handling on the server.

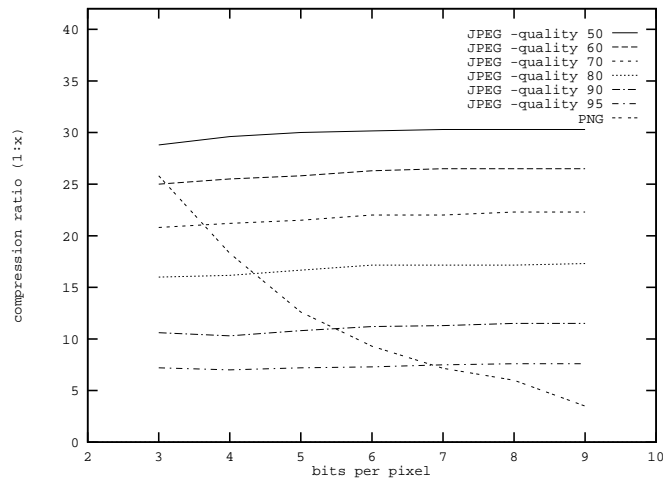
**Data reduction.** The adaptation of images to the client's display resources can lead to a significant decrease of the data volume. Figure 4 illustrates the process. Take, for instance, a 1280x1024 pixel true color image, which has a size of 3.75 Mbyte. Adapting it to a monochrome 320x200 pixel PDA screen leads to a data volume of 7.8 KByte, the adaptation of the same image to a 640x480, 256 color laptop display reduces the image data size to 300 KByte - without compression!



**Figure 4:** Image handling for different display types

Two steps are necessary to adapt the image to the client's display resources. First, the image size is reduced according to the requested or the client's default values. This can be done either by scaling (default) or by cropping if the client specifies a cropping rectangle. In a second step, the colors of the image are adjusted to fit the client's resources. This is currently done by octree [4] quantization and Floyd-Steinberg dithering [3]. Optionally, the image is adapted to the palette of the client.

**Compressor selection and configuration.** After the data reduction of the image, a compression algorithm is selected based on the color depth of the image, the achievable compression ratio or quality and the client request. Figure 4 illustrates the selection process for clients with different display capabilities. One interesting result of our tests has been, that JPEG is not always superior over the lossless(!) PNG image format [5], when compressing color images with a number of bit planes between 3 and 7 and a `-quality` parameter between 60 and 95. The dependency is shown in figure 5.



**Figure 5: PNG versus JPEG**

As this example shows, for the selection process knowledge about the properties of each compressor is necessary, which is stored in an extensible knowledge base. It contains the following facts for each compressor:

- suitability for images of color depth N
- lossy or lossless compression
- support of progressive refinement
- maximum achievable compression ratio
- configurability
- list for control parameter lookup, consisting of tuples of control parameter values, color depth values, compression ratios and image quality measures
- command line with placeholders for input file, output file and parameter value

Currently, the selection process consists of two steps:

1. From the list of available compressors, all algorithms are selected which are suitable for the color depth of the reduced image.
2. From the resulting reduced list, the algorithm is selected which meets all the criteria specified in the request. If no such algorithm exists, the one with the highest achievable ratio or quality is chosen.

If the selected compression algorithm is configurable, a configuration stage is necessary after the selection.

Parameter (-quality)	Compression ratio	Subjective image quality
30	1:40	20
40	1:35	30
50	1:30	40
60	1:27	50
70	1:22	60
80	1:17	70
90	1:11	80
95	1:7	95

**Table 2:** Configuration of the JPEG compressor

Either the desired compression ratio or the desired quality of the image can be requested by the client. Given the results from our compression algorithm investigation, we can now parameterize the compressor in such a way that the requested ratio or quality is achieved with a certain accuracy. This is done by looking up the nearest corresponding parameter value from the table or computing it by interpolation. Lookup supports the parameterization using more than one parameter, too. Table 2 shows the mapping between the parameter `-quality` of the JPEG compressor, the average compression ratio and the subjective image quality for 8 bit images.

The used JPEG implementation can be controlled by the parameter `-quality`, which, however, does not influence the image quality directly. Instead, it serves as a scaling factor for the quantization matrix. We found in our experiments that the `-quality` parameter does not exactly correspond to the subjectively perceived image quality. As many metrics for the term "quality" exist which may be highly dependent on the application domain, we chose to supply generic quality values in a range between 0 and 100 percent. The user of the protocol is free to define a mapping from his quality metric to the generic values.

Having decided which compression algorithm and which parameters to use, the parameters are passed to the compressor by replacing the placeholders in the command line by the current values. The compressors are considered to be external command line tools, which are called using operating system functions. Temporary files are used as a means for data exchange. This mechanism offers great flexibility.

## 4 Conclusions and Future Work

We presented the results of an investigation of the image compression algorithms

JPEG, wavelet and fractal compression as the basis for a quality and resource controlled image transmission protocol. Since JPEG was found to be the fastest and most stable algorithm, we selected it for use in the image transmission protocol, which we proposed in the second section of this paper. This protocol considers quality demands and client resources. In a two-step process, the requested image is reduced to fit the client's display resources and compressed using an image compression algorithm that has been selected and configured according to the clients quality or data volume requirements. An extensible knowledge base supports the incorporation of new image compressors.

Some problems remain to be solved and need further work:

- The quality judgement of the compressed images should be repeated with a greater number of subjects.
- If not all criteria are met, the current selection algorithm returns the compressor with the highest achievable compression ratio. This has to be changed to return the compressor which meets most of the criteria. Assigning weights to the criteria can help resolving in conflicts.
- Support for connection breakdown handling should be added.
- An image compressor which creates thumbnail catalogues to support visual search operations (as described in [2]) should be integrated. This includes the extension of the request structure to support lists of image IDs in one request.
- A generalized handling of data reduction and data compression modules is needed to support, e.g., alternative dithering or scaling methods.

## Acknowledgement

This work is part of the MOV1 Project supported by the German Science Foundation (DFG) under contract no. Schu 887/3-1.

## References

1. Lu, J.; Algazi, V.R. and Estes, R.R.: Comparison of Wavelet Coders Using the Picture Quality Scale (PQS), invited paper, in: Szu, H.H. (ed.): Wavelet Applications II, Proc. SPIE, Vol. 2491, pp. 1119-1130, April 1995.
2. Burton, C.A.; Johnston, L.J.; and Sonenberg, E.A.: An Empirical Investigation of Thumbnail Image Recognition, in: Proceedings Information Visualization 95, October 30-31, 1995, Atlanta, GA, pp. 115-121

3. Floyd, R. and Steinberg, L.: An Adaptive Algorithm for Spatial Gray Scale, in: Society for Information Display 1975 Symposium Technical Papers, 1975, 36
4. Gervautz, M. and Purgathofer, W.: A Simple Method for Color Quantization: Octree Quantization, in: Magnenat-Thalmann, N. and Thalmann, D. (eds.): New Trends in Computer Graphics, Springer, Berlin, 1988, pp.219-231
5. Crocker, L.D.: PNG: The Portable Network Graphic Format, in: Dr. Dobb's Journal, Vol. 20, July 1995
6. Jaquin A.: Image Coding Based on a Fractal Theory of Iterated Contractive Function Systems, SPIE Vol. 1360, Visual Communications and Image Processing 1990
7. Hilton Michael L., Jawerth Björn D., Scngupta Ayan, Compressing Still and Moving Images with Wavelets, Multimedia Systems, Vol. 2, No. 3, 1994
8. CCITT Draft Recommendation T.82, ISO/IEC Draft International Standard 11544 Coded Representation of Picture and Audio Information - Progressive Bi-Level Image Compression, 1992
9. Fischer, Y. (ed): Fractal Image Compression: Theory and Application, Springer, New York, 1995
10. Pennebaker, W.B. and Mitchell, J.L.: JPEG Still Image Data Compression Standard, Van Nostrand Reinhold, New York, 1993.