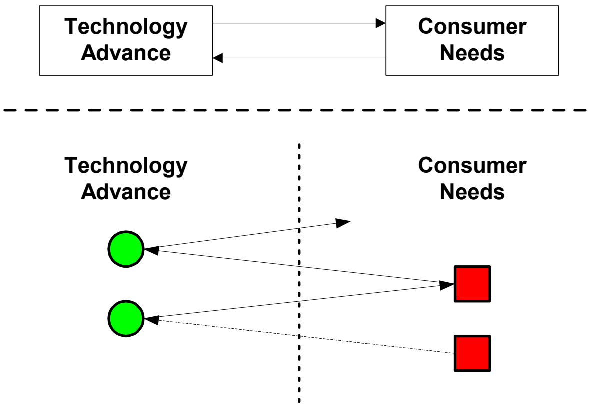
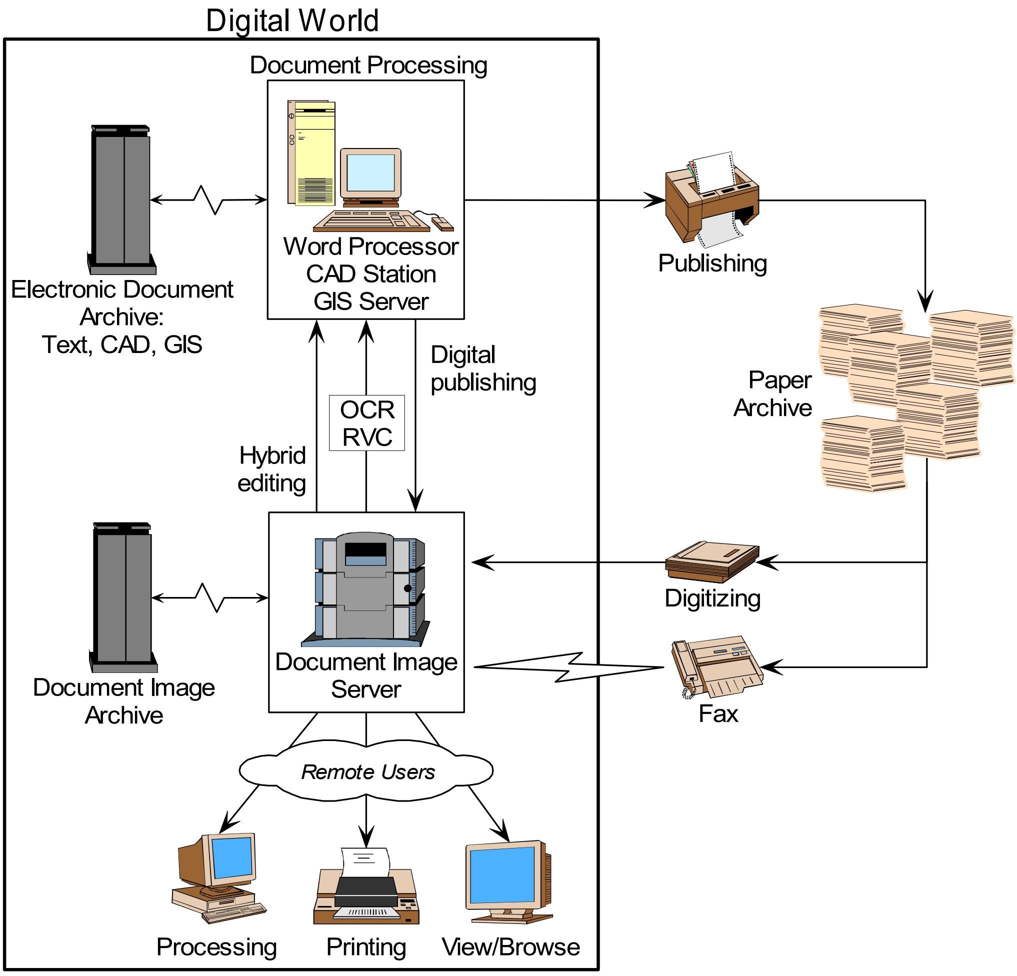
Real-time imaging in mobile environment.

Dr. Eugene Ageenko. CS Dept., University of Joensuu, Ageenko@cs.joensuu.fi

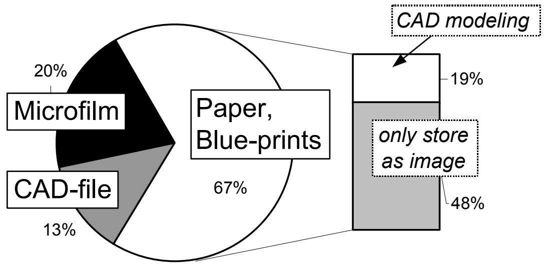
The technical world is developing in the constant process: consumer needs stimulate research and development work, whereas new advances in technology bring to the user new abilities and create new problems and needs. Nobody knows how humans have made their first technological advance (was it fire?). The scientists believe it was in accident, however this accident was awaited for a long time, and come exactly when it was needed.



The real-time imaging development has been started yet in the latest eighties – earlier nineties. At about this time, the world overloaded by the waste of paper documents decides to go digital, and convert documents existing mostly in the paper form into a digital format.

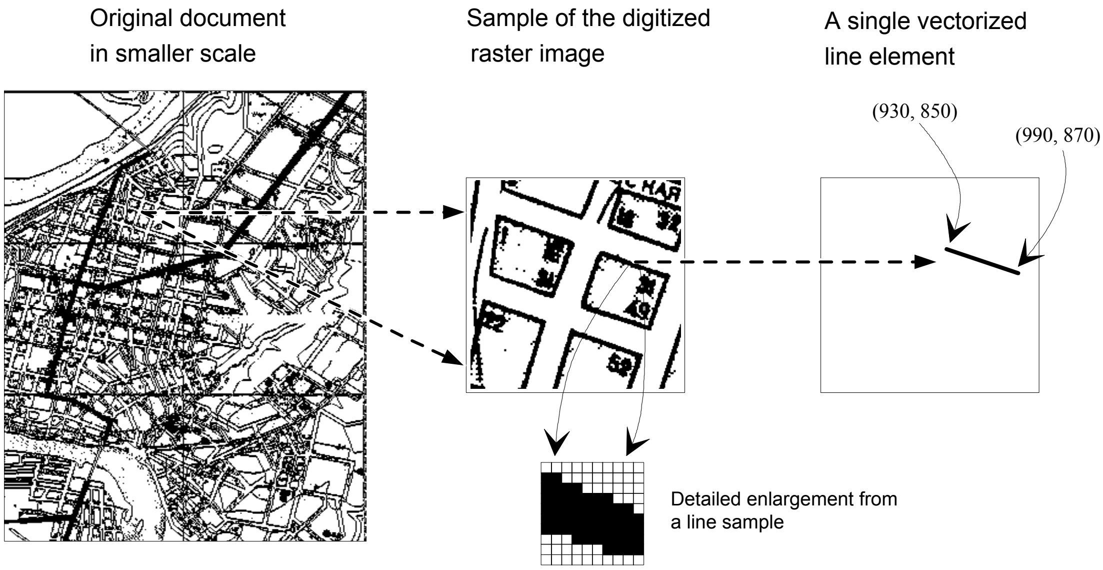


The amount of paper documents in the world has been estimated to exceed 10^{12} pieces (1998, O.A.S.), and this quantity has been estimated to double every three years. In particular, only in U.S. there were estimated to have about 8B engineering drawings, of which more than enough amount existed on the paper. The expenses emerged from document losses reached nearly 10 % of all document management expenses.

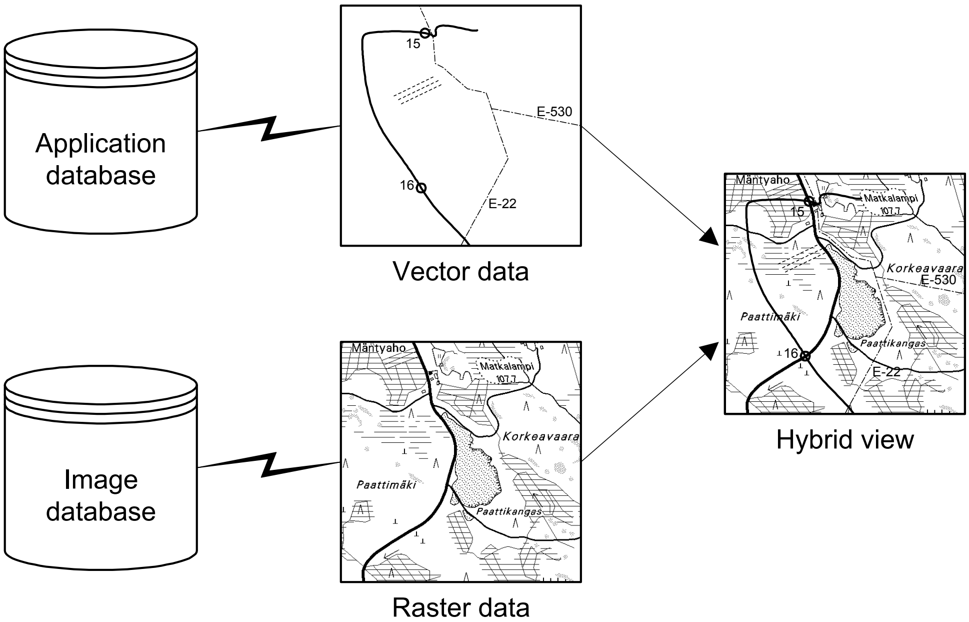


The microfilm was an existing alternative solution, however digital documents had important advantages. In particular, they can be stored more economically, searched easily, reproduced without loss of quality, and transmitted quickly.

Existing at those time technologies did not permit a high quality conversion of printed and drawn documents back into original textual or CAD format by the means of OCR or Raster-to-Vector Conversion (see picture below). However, the scanning process itself became more cheap and popular, and scanning resolution have reached 300-400 DPI, which was in 1993 considered **legal** to substitute document images for original papers in storing and accessing information.



The further investigations have also shown that a strong requirement for conversion of the raster images into the original vector (CAD) form is exists only for parametric modeling and control system applications. These applications represent less than 15 % of all applications where engineering documents are used [Wils96]. In most other applications, the raster format is often sufficient, especially if *hybrid editing* is supported. Hybrid editing permits using both raster and vector data simultaneously and benefit from each other.

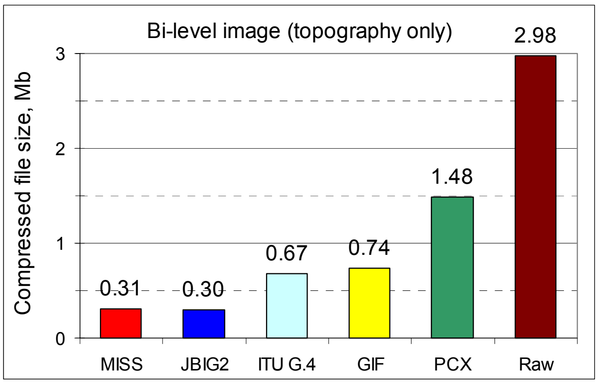


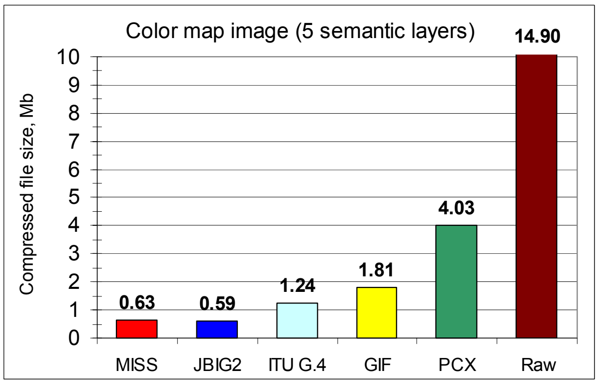
Moreover, using state-of-the-art image compression techniques already developed in 90s it would be possible to achieve the same degree of compaction of document images as if using documents in the original CAD and Word Processor or PostScript format.

Making a flashback into the past, the processing power of the computers in early nineties were about the same level as of today mobile and portable devices:

	Early 90s	Mid 90s
Processor speed	1-10 MIPS	50-100 MIPS
RAM	< 1 Mb	16-32 Mb
Storage space (HD)	< 4 Mb	~ 100 Mb
Display size	320×200	640×480
Network speed	1.2 Kb/s	14.4 – 56 Kb/s

To understand how the compression is important for storing image data, we draw the following example. A typical topographic image representing a single map-sheet will have dimensions of ~ 5000×5000 pixels. The following diagrams show the sizes of the image file in the raw and compressed form for the black-n-white and color representations. The JBIG2 and MISS columns on the picture correspond to the latest state-of-the-art technique (JBIG2) and our innovation (MISS).





One can think about the compression as consisting of two components: modeling and coding. The modeling is the way to represent the data, when the coding is in assigning the codes to the original message (image) according to the model. The existing image compression technology has not delivered sufficient compression rates to fit large enough images into a small storage space. The methods were lousy mostly because they were based on the 1-dimensional models and have used integral codes (that assign integral number of bits per code).

Saying more, there were a nice coding technique, namely the arithmetic coding, developed yet in 1979 by Rissanen and Langdon (IBM RC), and concluded to international standard 11544 (also ITU/T Rec. T.82), namely JBIG. JBIG has been supposed to replace ITU G.3 coding technique used in all the existing fax machines by the end of 20th century. (Did it? Did you hear about JBIG?) However at that time, there were neither engine nor the processor to be able to perform arithmetic coding in the real-time.

A second major drawback of existing compression techniques (including JBIG), is that entire image (at least prior to the point of interest) must be decompressed in the memory (means also entirely transmitted to the client location) before it can be presented to the user. Using supposition that client device may not have sufficient resources to hold entirely decoded image, and high-sped channel (and fast processor) is not available to transmit (decode) the entire image is real-time, the operations with remotely located images would be remote (read, **impossible)**.

The user needs made us to develop the requirements for the Real-time Imaging Storage System:

1. High compression rates – improves transmission and saves on storage (already possible with JBIG and JBIG2).
2. Fast decoding – real-time operations.
3. Instant preview property – quickly extract the thumbnail image.
4. **Spatial access** – be able to operate directly on the compressed stream, decode **on the fly**!

As we have noted, typical viewing devices have smaller size and resolution than the original raster image and thus, only a small fragment of the entire image may be viewed at a time. If spatial access is supported, an image may be interactively browsed on the viewing device. When the image is scrolled, a new part of data is retrieved and decompressed on the fly. In this way, spatial access eliminates time delays caused by image decompression and transfer. The thumbnail image may serve as a map to locate the desired part of the image at a higher scale.

The storage system (*Document Image Storage System*) satisfying the above requirements has been developed in 1998-2000. Referring to development spiral, this technology gave us the possibility to extend into the Cartography Imaging area and develop the *Map Image Storage System* – MISS ©.

The system supports the following properties:

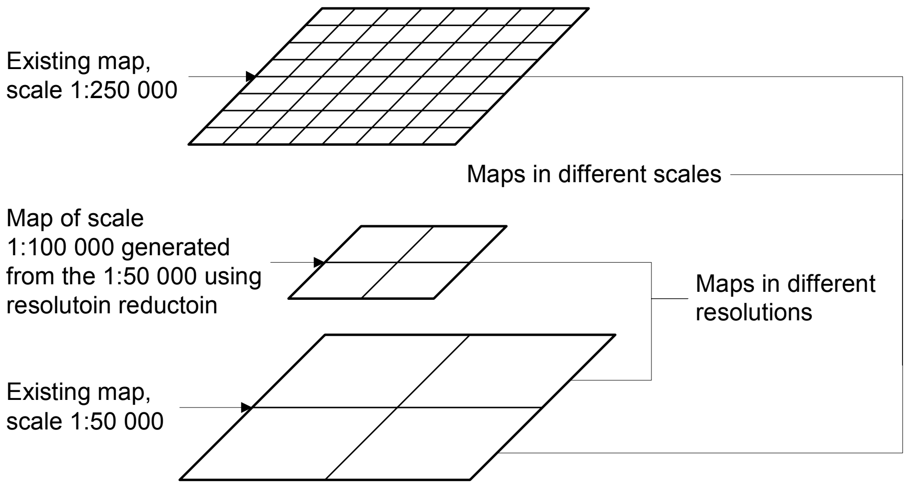
1. Compact storage size (state of the art);
2. Multi-scale representation (zooming);
3. Fast scrolling (panning) ability, via the spatial access feature.

The original MISS images are stored in the server-side database. Spatial views are generated for the client-side application using compressed raster image format organized so that it supports the zooming and panning requirements. In this way, raster format is suitable in applications, where the maps are needed for viewing purposes only.

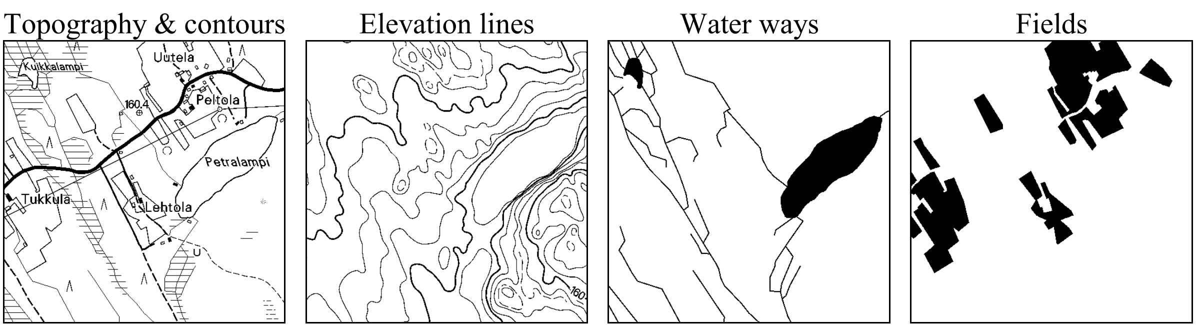
Furthermore, the system does not depend on any database or vector format as digitized raster maps can be easily generated are reproduced from any source format, including paper maps. Another advantage of the system is that it requires only a modest memory and computing resources in order to be operational in real-time environment.

The general idea of the MISS is in the separation of the image into the:

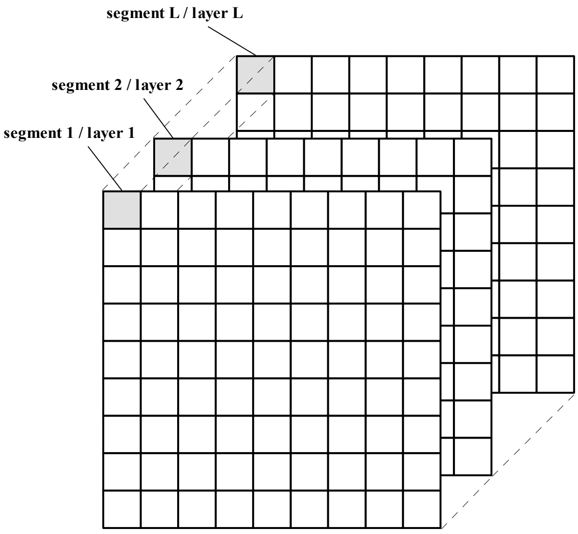
- multiple scales:



- semantic layers:

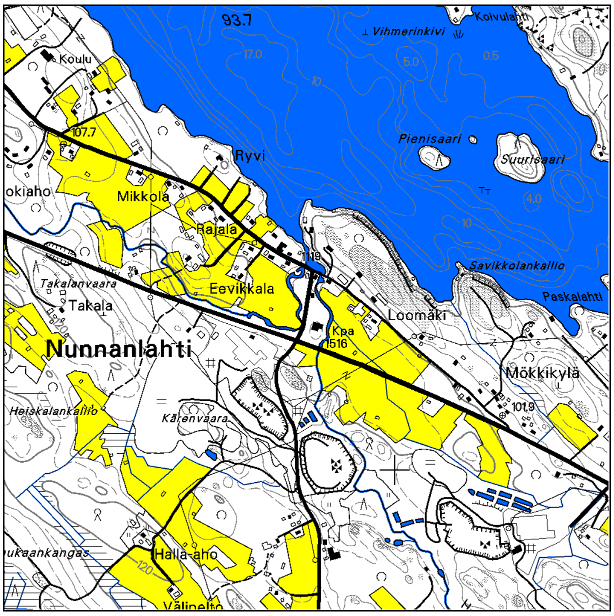


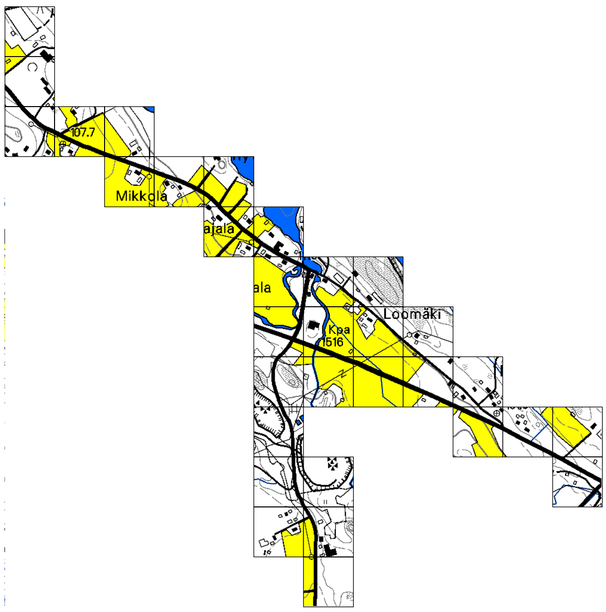
- block tiles:



and in the use of the special encoding schemes satisfying with the above sub-division and system requirements.

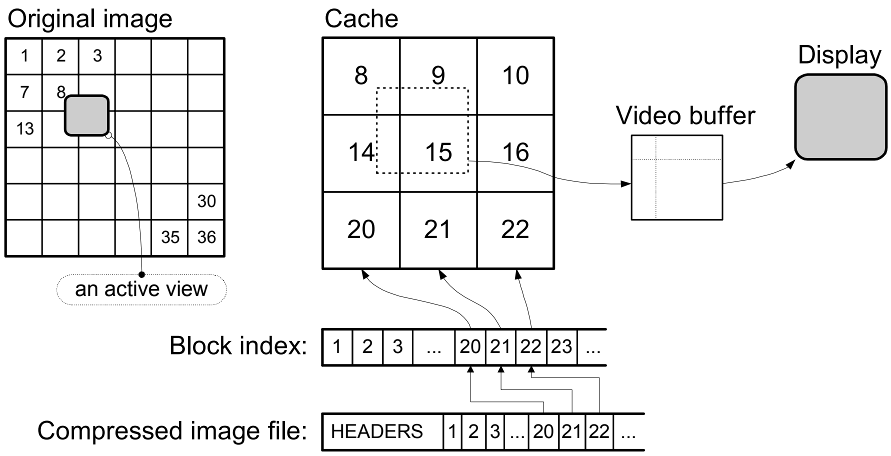
The following picture shows how the map image is operated in the client device. The map blocks along the client's route (providing that the client's location is dynamically determined used either of positioning services) are dynamically decoded (or retrieved and decoded) and used to create a picture shown on the device display.





A following picture illustrated these operations in more details. First, the file possessing map image of the desired spatial location and scale is accessed and its header part is retrieved and parsed in the memory. From the headers, the image type and structure are determined and block index table is built. The table indicates the size and locations in the file of the compressed image blocks. All supplementary data required for image decoding (such as initial model, etc) is also retrieved and kept in memory until this map image is no longer used.

Next, depending on the requested location, the system calculates the map image fragment, which needs to be displayed. The blocks covering this fragment are retrieved and decoded. If the map image is accessed remotely, the retrieved blocks are also stored (in the compressed form) in a local storage space for further use. When the position of the object changes, the view is updated by decoding new image blocks in the direction of movement.



Bringing all the features together, there are four possible scenarios for the use of MISS in the mobile environment:

1. Client-side approach (demonstration):

    - *Laptop*: compressed images are stored on the CD (eg. Tiekartasto CD), full capabilities, positioning using external GPS (GPS card).

    - *PDA*: compressed images are stored or preloaded using a bigger computer and faster Internet connection (possible loading over the air in a single bundle).

    - *Dedicated client* (Benefon Esc!): compressed images are preloaded.

2. Server-side approach (http://cs.joensuu.fi/~ageenko/research/edm_demo.html):

    - *Thin client + WAP*: image is prepared (decompressed) on the server, only current view is served to the client in a compatible image form; positioning using MPS or manual entry.

    - *JavaME client*: additional features for improved interactivity, eg. smooth map panning, etc.

Dynamic map loading scenarios:

3. Client-server approach, shift on the server:

    - *Client, capable for image decoding and HTTP communications* (advanced phone, eg. Nokia Communicator): if the compressed map file existing on the device (either preloaded or saved from previous trip) does not cover the requested fragment, the client will request the data from the server. Server extracts the data for the requested fragment and serves it back in the original (compressed) form. Client adds this data to the existing compressed file, and proceeds as if the data is locally present.

4. Client-server approach, shift on the client:

    - *Client, capable for image decoding, HTTP communications and with sufficient (or interchangeable) memory resources* (PDA, Laptop, etc): as in the previous scheme, however the client has preloaded (or the ability to load over the air) a map index table (can be large), which will be used for requesting from the server the compressed file parts corresponding to the desired image fragments. In this case the server is a simple HTTP server.

References

[1] M.-J. Kraak, et al, *Cartography : Visualization of Spatial Data*, Addison-Wesley, 1996.

[2] M.-J. Kraak, A. Brown, *Web Cartography*, Taylor & Francis, 2000.

[3] E.D. Kaplan (ed.), *Understanding GPS: Principles and Applications*, Artech House Telecommunications Library, March 1996.

[4] S. Dye, S. Buckingham, *Mobile Positioning*, Mobile Lifestreams, 1999.

[5] E.A. Fox, et al. (eds.). Digital Libraries. [*Special issue of*] *Communications of the ACM* **38** (4), 1995.

[6] H. Samet, *Applications of Spatial Data Structures: Computer Graphics, Image Processing, GIS*. MA: Addison-Wesley, Reading, 1989.

[7] GeoData, *CD-tiekartasto Suomi 1:250* 000 (*CD Roadmap Catalog Finland*), WSOY, Helsinki, 1999. (in Finnish)

[8] H.W. Gellersen, P.J. Thomas (eds.), *Proc. Second International Symposium on Handheld and Ubiquitous Computing* (HUC'2000), Bristol, UK, Springer Verlag, September 25-27, 2000.

[9] R.B. Arps, T.K. Truong "Comparison of international standards for lossless still image compression," *Proceedings of the IEEE*, **82** (6), 889-899, June 1994.

[10] J.D. Murray, W. vanRyper, D. Russell (ed), *Encyclopedia of Graphics File Formats*, 2-nd ed, O'Reilly Associates Inc, 1996.

[11] B.G. Haskell et al., "Image and video coding – emerging standards and beyond," *IEEE Trans. Circuits and Systems for Video Technology*, **8** (7), 1998, 814-837.

[12] ISO/IEC, *Progressive Bi-level Image Compression*, 11544, ITU-T Recommendation T.82, 1993.

[13] ISO/IEC, *Final committee draft for ISO/IEC International Standard 14492*, 1999. (`http://www.jpeg.org/public/jbigpt2.htm`)

[14] P.G. Howard, F. Kossentini, B. Martins, S. Forchammer and W.J. Rucklidge, "The emerging JBIG2 standard," *IEEE Trans. Circuits and Systems for Video Technology*, **8** (7), 838-848, 1998.

[15] G.G. Langdon, J. Rissanen, "Compression of black-white images with arithmetic coding", *IEEE Trans. Communications*, **29** (6), 858-867, June 1981.

[16] W.B. Pennebaker, J.L. Mitchell, G.G. Langdon, R.B. Arps, "An overview of the basic principles of the Q-coder adaptive binary arithmetic coder," *IBM Journal of Research and Development*, **32** (6), 717-726, 1988.

[17] D. Tompkins, F. Kossentini, "Additional Extension Segments for JBIG2," ISO/IEC JTC 1/SC 29/WG1 (ITU-T SG8), Document No 1318, July 1999.

[18] W.B. Pennebaker, J.L. Mitchell. *JPEG Still Image Data Compression Standard*. Van Nostrand Reinhold, New York, 1993.

[19] M.J. Weinberger, J. Rissanen, R. Arps, "Application of universal context modeling to lossless compression of gray-scale images," *IEEE Trans. Image Processing*, **5** (4), 575-586, April 1996.

[20] E.I. Ageenko and P. Fränti, "Enhanced JBIG-based compression for satisfying objectives of engineering document management system", *Optical Engineering*, **37** (5), 1530-1538, May 1998.

[21] E.I. Ageenko, P. Fränti, "Compression of large binary images in digital spatial libraries", *Computer & Graphics*, **24** (1), 91-98, 2000.

[22] E.I. Ageenko, P. Fränti, "Forward-adaptive method for compressing large binary images," *Software Practice & Experience*, **29** (11), 943-952, 2000.

[23] E.I. Ageenko, P. Kopylov, P. Fränti**,** "Optimizing context template for compression of multi-component map images," *Proc. GraphiCon'00*, Moscow, Russia, pp. 151-156, 2000.

[24] J. Miano, *Compressed Image File Formats: JPEG, PNG, GIF, XBM, BMP*, (ACM Press), Addison-Wesley , Boston, 1999.

[25] T. Welch, "A technique for high-performance data compression", *IEEE Computer*, **17** (6), 8-19, June 1984.

[26] J. Ziv and A. Lempel, "Compression of individual sequences via variable-rate coding", *IEEE Trans. Information Theory*, **24**, 530-536, 1978.

[27] G. Roelofs, *PNG: The Definitive Guide*, O'Reilly & Associates, Cambirdge, MA, June 1999.

[28] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression", *IEEE Trans. Information Theory*, **23**, 337-343, 1977.

[29] National Land Survey of Finland, Opastinsilta 12 C, P.O.Box 84, 00521 Helsinki, Finland. (`http://www.nls.fi/index_e.html`)