

# A Fast Storage Unit With Integrated Database Processor for Image Data

Lieckfeldt, P. ; Missling, K.-D. ; Neumann, B.

Academy of Sciences of the GDR

Institute for Space Research

Satellite Ground Station

Kalkhorstweg

Neustrelitz

2080

## Abstract

This paper gives a survey of an efficient image storage system with integrated database functions.

Storing the physical image data on external media causes mainly two problems: storage capacity and access time. An intelligent disk controller for disk arrays which tackles both problems will be presented.

The management of images can be done in easy cases with a conventional DBMS. Extending such a DBMS by a specialized search hardware to a DB Machine allows a faster image retrieval. A higher level of image retrieval demands a semantic description of the image content.

## 1. Introduction

Images are a very important source of information in many fields (e. g. remote sensing, meteorology). In nearly all cases images and related information must be stored and managed. A concept for an Image Database Management System (IDBMS) has to offer solutions for the following problems:

- On which media the huge amount of data (e.g. series of raster data, maps) should be stored? How is it possible to guarantee a required data transfer rate?
- How is it possible to access to the image information fast and in a user-friendly manner?

For applications with extreme time requirements we conceived an image storage unit with high storage capacity and data transfer rate (in MByte/s range). The image data management is done by an integrated IDBMS which allows a very quick access to image annotations. In chapters 2 and 3 a detailed description of both components (storage unit, IDBMS) is given.

## 2. Storing of physical images

If short I/O-service time is a goal implementing an IDB then direct access storage devices (DASD) as hard disks or optical disks are often used as storage medium. They have a sufficient cost per bit ratio, an access time in the millisecond range and an acceptable storage security. The data transfer rate is in the range of some MByte/sec.

If demands for retrieval are very high the so called "access gap" between primary memory and disks limits the system performance. Because there isn't any new stor-

age technology to overcome this limitation it's worth while to implement systems in disk technology with better properties. There are two main ways to improve these properties.

The extremely raise of the access and transfer quality of the devices results to high performance memories, for example IBM 3380 model AK4. This way is connected with growing technological difficulties and exponential increasing costs.

The other way joins the knowledge, that technological limitations can be by-passed on the architecture level, with the fact, that the development on the pc-market produce inexpensive disks, which follow the high end disks in rapid succession.

If it is possible to organize an array of inexpensive disks as one unique device a new level of access behaviour can be reached by low costs. A theoretical comparison of the IBM 3380 model AK4 with an array of 100 3.5 inch pc-drives Connors CP3100 is shown in figure 1.

A needed array controller not only will satisfy higher demands for organisation and control of I/O requests.

If logical connected data items spreaded over all drives of the array the MTTF is :

$$MTTF_{array} = MTTF_{single\ disk} / array\ size$$

That means, already arrays of small size haven't a sufficient reliability. The reliability of the array must be improved by using redundant information. The volume for redundant data can be little

in practice. The enormous reliability shown in fig.1 can be realized by 10% of check information. So the controller must include parts to generate needed redundant information.

With such a controller it is possible to accomplish not only a cheap and powerful storage device but also a device with a very high reliability usually without backup.

	Unit	High-performance drive IBM 3380 AK4	Array of 100 inexpensive disks
capacity	MByte	7500	10000
data rate	MByte/s	12	100
I/O-rate	I/O /s	20	3000
power	kW	7	1
cost	\$	100000	100000
cost per MByte	\$	13.33	10
volume	cub.feet	24	10
reliability	h	30000	820000

Fig. 1: Comparison of SLED with RAID Level 5 /1/

Figure 2 shows the basic concept of such an array controller. The controller computer receives orders from the host. It generates partial tasks for the other components of the controller, supervises and synchronises their work and apportions them data from the total data volume.

A possibility of data partitioning would be into small components (Bit or Byte divided). That results in a high data transfer rate and a short response time for big amounts of data (images and sequences of images). But if a large number of small transfers are ordered seek time and rotational latency dominate in the response time of the array. That leads to growing response time in practice.

A better method dividing data is into components match some physical sectors of the DASD. Then a parallel work of some single disk controller on different transfer tasks is possible. Every single disk controller services only one disk, stores intermediately the data, detects and corrects soft errors of the data stream. Instead of it an intermediate memory pertains to the single controller.

The correction hardware generates additional information to the input data stream, needed for prospective reconstructions and restores destroyed data. It is possible to store the needed redundant information on special disks, called check disks. Due to several reasons these check disks are a bottleneck and deteriorate the system performance. Therefore in our project we distribute the check data across all physical disks. Then the disk array is a level 5 RAID system from the classification of [1].

The connection component links all controller parts. It can be implemented by a bus. But there is the disadvantage that such a structure couldn't fit to the needs of the actual transfer tasks.

In the first step of implementation we will build up a controller for 16 disks connected by a modified bus structure. It should work with a reliability of 60,000 h. Performance estimations show a transfer rate for large amounts of data from 10 MByte / sec and a I/O rate of 300 I/O per second.

### 3. Image Database Management System

The term "Image Data Base" (IDB) is often used with several meanings. The most simple IDB is only a collection of images (image archive) - the question how to access to these images is not solved. In this sense the above described storage complex with high capacity is a well suited device for installing such a simple IDB. It should be mentioned that sometimes it is possible to use "intelligent" data distribution methods to reduce sub-image retrieval time [2].

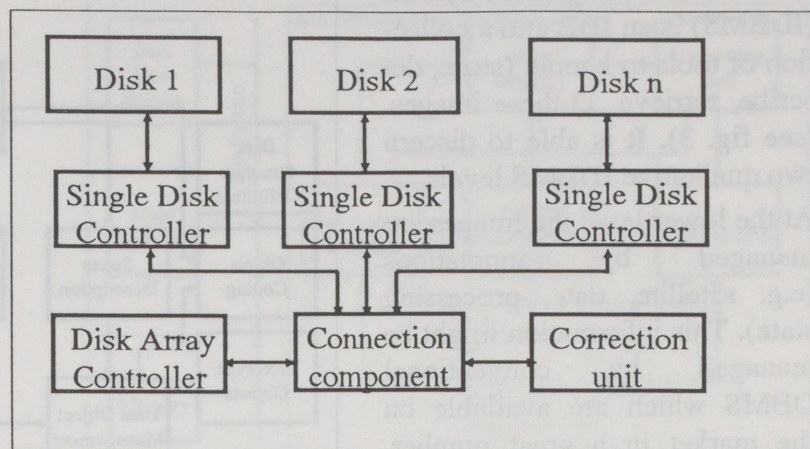


Fig. 2: Disk Array Controller Configuration

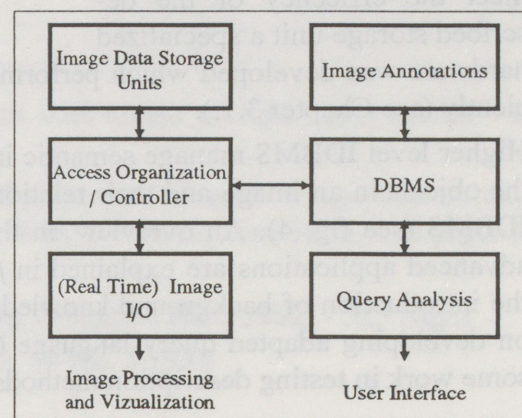


Fig. 3: Standard IDBMS Configuration

A true IDB Management System (IDBMS) is an IDB and a collection of tools to handle (store, describe, retrieve ...) these images. (see fig. 3). It is able to discern two qualitative IDBMS levels.

At the lower level the images are managed by annotations (e.g. satellite, date, processing state). This information might be managed by conventional DBMS which are available on the market in a great number. Such low level IDBMS fulfil the functional requirements for many applications using image-like information but the speed of information retrieval is not always satisfactory. In order to meet the efficiency of the described storage unit a specialized hardware was developed which performs searching operations in DBMS's very efficiently (see Chapter 3.1.).

Higher level IDBMS manage semantic image information. The geometric structure of the objects in an image and their relations (e.g. neighbourhood) are stored in such an IDBMS (see fig. 4). An overview on this rather complex task may be found in /3/; advanced applications are explained in /3,4/. Research activities are concentrated on the introduction of background knowledge in the process of scene understanding and on developing adapted query language (Image Query Languages). At present we do some work in testing description methods well suited for non- exact object matching.

### 3.1. Main Memory DBMS with hardware accelerator

A relational DBMS was conceived based on the hardware architecture shown in fig. 5. A detailed description can be found in /5/.

Due to the intended applications of the whole system the DBMS was conceived as an Main Memory DBMS, i.e. that all annotations are kept in the main memory during the working phase of the system. The external memory serves only as a backup medium. Main advantages are the fast data access and the independence of physical data distribution from the user access behaviour.

The most important algebraic operator in a relational DBMS is SELECT. Presumed the data in a MMDBMS are organized in a sequential manner it is possible to construct a rather simple but very efficient hardware to perform the SELECT operator. A simulation shows that such a functionally adapted hardware is faster by a factor 2-20 compared to a pure software solution using well known index structures (this statement must be not true for hash support) /6/.

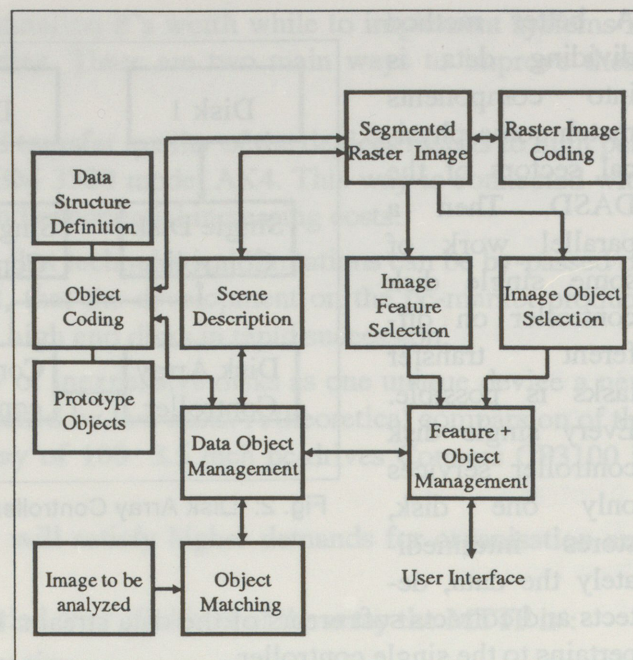


Fig. 4: IDBMS schema using semantic features

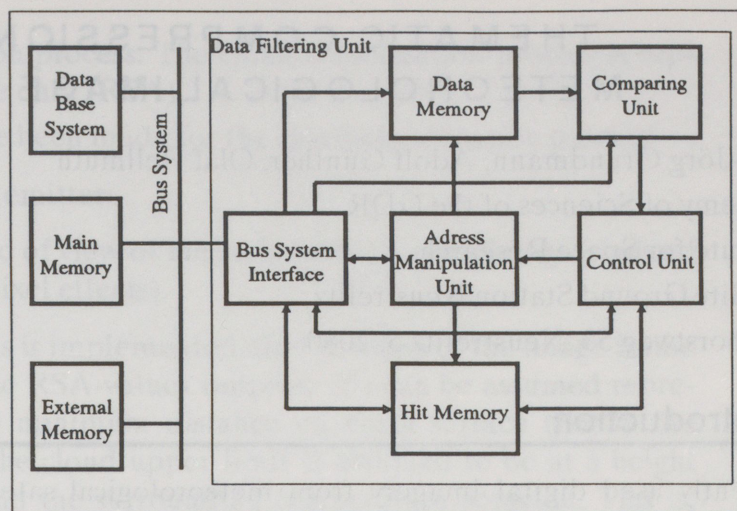


Fig. 5: Data Filtering Unit Configuration

### Literature

- /1/ Mann, T. : Breaking the data rate logjam with arrays of small disk drives. - In: Electronics, February 1989. - pp. 97-100
- /2/ Lieckfeldt, P. ; Neumann, B. : Optimized allocation of raster image data on magnetic disk devices. - Internal Report Satellite Ground Station, June 1986. - 18 pages (in German)
- /3/ Ballard, D. ; Brown, C. : Computer Vision. - Prentice Hall, 1982. - 523 pages
- /4/ IEEE Trans. on Software Engineering Vol. 14 No. 5 1988
- /5/ Patent DDR G06F/314 810 8
- /6/ Neumann, B. : Investigations on data filter time requirements. - Internal Report Satellite Ground Station, June 1989. - 9 pages (in German)