

RasDaMan: Raster Data Management in Databases

P. Baumann¹, P. Vivas², and A. Martínez³

RasDaMan is a system for raster data management being developed within the ESPRIT Long Term Research project 20073 (Raster Data Management in Databases), funded by the Directorate General III of the European Commission. The project focuses on developing comprehensive database support for raster data or *multidimensional discrete data* (MDD) and will be trialed by developing a geographic information system application. Whereas the most prominent examples of raster data still are images, i.e., 2-D matrices, raster data are not restricted to the 2-D case – in principle, any natural phenomenon becomes spatio-temporal raster data of some specific dimensionality once it is sampled and discretized for storage and manipulation in a computer system. Multidimensional database technology involves DBMS support wrt. modelling of arrays with arbitrary size, dimension, and base type; a declarative query language suitable for such arrays; adapted storage management, including tertiary storage; and storage and retrieval optimization, in particular wrt. disk access and network traffic.

Following a deep investigation, it can be stated that current systems either do not offer modeling flexibility and data independence on raster data to a degree comparable with the classical database types, or they do not complement such concepts with storage mechanisms suited for huge arrays. On the other hand, raster DBMS support indeed is indispensable for a host of applications in widely spread areas – office applications, CAD image management, remote sensing, environmental planning and control, medical imaging, historical and geographic information systems, OLAP⁴/data mining tasks, and scientific visualization, to name but a few, require database support to an increasing degree. In fact, any by nature analog phenomenon finally appears as discrete data of some specific dimensionality when sampled by some sensor and fed into an information processing system. The main characteristic of such data is that they form regular d-dimensional arrays which frequently are too big to fit into main memory as a whole. We call such structures *multidimensional discrete data* (MDD) since they are obtained via a regular sampling (or discretization) process of one or more properties over a multidimensional spatial domain. For example, an image and a sound sequence are examples of two-dimensional and one-dimensional MDD objects, respectively.

¹ FORWISS. Germany (baumann@forwiss.tu-muenchen.de)

² Centro Nacional de Información Geográfica. Spain (PVIVAS@geo.ign.es)

³ STI, SA. Spain (sysnet@bitmailer.net)

⁴ *Online Analytical Processing* – the art of arranging business data in multidimensional data cubes to gain new insights through operations like projection, consolidation, and statistical analysis.

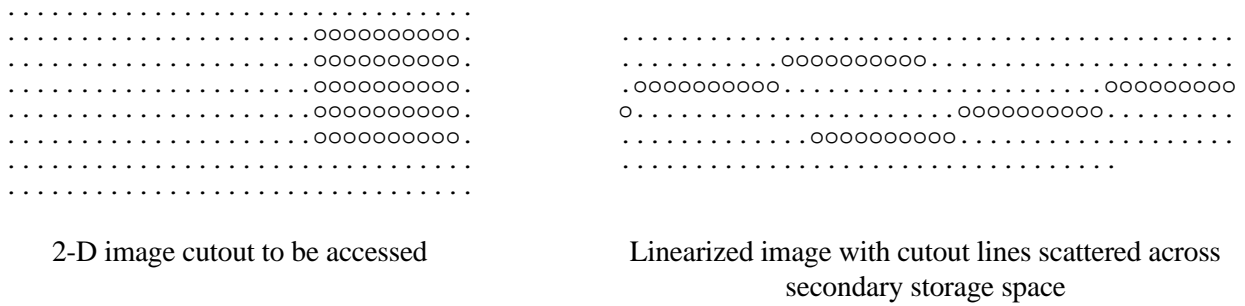


Figure 1: Mass storage linearization effects

Therefore, we claim that a DBMS must offer comprehensive MDD management facilities. On the conceptual level, this means definition of *arrays over arbitrary pixel types*, not just a predefined selection of pixel types like integer and real. A *coherent, orthogonal set of operations* must be available on such array types which allows for enough modeling flexibility to express MDD retrieval and manipulation. The physical database layer must support the array concept by providing *efficient access methods* for n-dimensional arrays of basically arbitrary size. Concealment of these internal storage structures, i.e., *data independence*, is an essential prerequisite for cooperative, open multimedia environments distributed over heterogeneous networks, as only then image structures can be transmitted and reassembled appropriately according to the target machine's representation needs.

Please note that we do not demand to embed a fully-fledged imaging system into the database machine - this is an inadequate solution anyway, but first of all contradicts modularity. To stress our introductory example again: Certainly nobody will perform number crunching tasks in a database system; yet it is feasible not just to store and load real numbers, but to use them in query expressions, too.

The previous arguments have been driven by an application-oriented point of view. Let us now view raster data management from a scientific position. In the discipline of visualization, where the areas of computer graphics, image processing, computer vision, computer-aided design, signal processing, and user interface studies converge into one unifying framework for the processing of visual information [McC0-87], several representations of a scenery (an image in its most general meaning) are distinguished. Krömker [Kröm-91] proposes a reference model for visualization which is particular suitable for database investigations because classification is done along the data structures on hand (Figure 2). Three of the six layers introduced in this reference model are relevant for DBMSs dealing with visualization structures, i.e., spatial DBMSs [Baum-93]:

- The *Symbolic Representation Layer* knows about abstract scene descriptions, however without an explicit description of geometry and properties of the entities modeled.
Example: A 3-D scenery consisting of a house with a tree next to it might be described through the entities `House` and `Tree` with a relationship `is-north-of` between them. `House` could have attributes like `#Floors` indicating the number of levels or `address` for its postal address.
- The *Geometry/Feature Layer* covers geometric descriptions, appearance properties, and viewing parameters. Vector graphics comprises a subset of such data structures.
Example: On this level, the house/tree scenery is described without the specific semantics of a "house" and a "tree", but with information about sizes, geometric locations, appearance etc. Thus, `House` in this view consists of (i.e., is bounded by) 2-D regions positioned in Euclidean space, the walls and its roof having assigned individual surface properties like color and roughness. A complete scenery description additionally requires one or more light sources with attributes color, intensity, and location.

- On the *Digital Pixel Layer*, a scenery is discretized in both space and color, yielding a raster image. A raster image consists of a finite set of points in the discrete coordinate space Z^d where each point has some value, its color, associated.

Example: Adopting a specific point of view and viewing angle of the observer/camera as well as a certain pixel resolution and color space, the geometric scenery can be rendered yielding a raster image of house and tree.

From the point of view of conceptual modeling in databases, the Symbolic Layer is covered by semantic nets; the Geometry/Feature Layer is tackled by vector databases (usually named geo databases), and entities belonging to the Digital Pixel Layer are maintained in image databases.

Let us elaborate on the difference between vector and raster representation of spatial data. It is important to realize that these representations are not only very different in terms of structures and operations, but indeed comprise substantially different kinds of information about one and the same real-world entity. This becomes evident when we look at the means for transformation between both representations.

Rendering a (2-D or 3-D) geometric model generates a raster image which depends on various parameters usually not captured in the geometry/feature information, e.g., the curve technique employed [Mort-85] and the illumination model implemented in the renderer [Bouk-70, Gour-71, Phon-75]. And, of course, discretization of geometric elements implies a loss of geometric accuracy and structure information (e.g., does a long dark shape in a raster image represent a line or a thin region?).

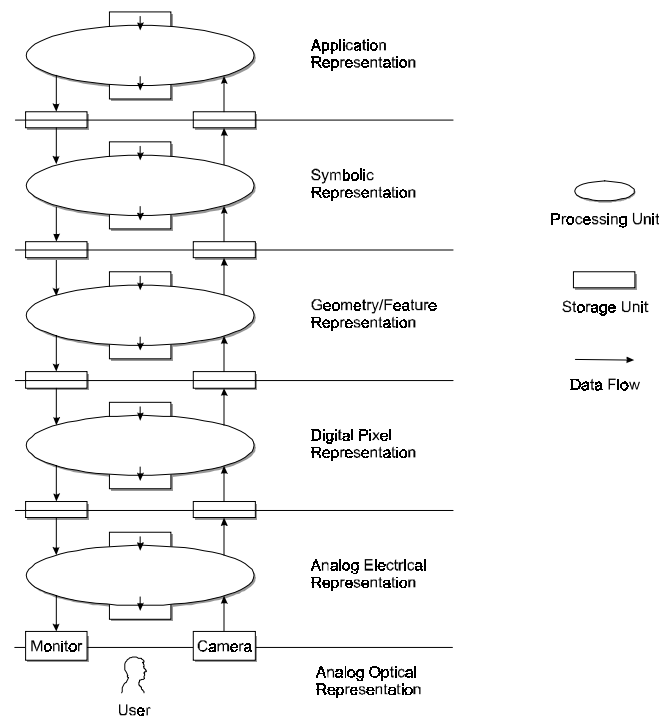


Figure 2: Krömker's reference model for visualization

Feature extraction methods try to interpret raster images to recognize points, lines, and regions which are assumed to be encoded in the pixels. This works acceptably well in small universes of discourse, e.g., for technical drawings with their highly stylized graphical vocabulary. However, there is no algorithm which performs reasonably well on any kind of image and under all circumstances; most of all, images frequently contain information which cannot be cast into points, lines, and regions bounded by lines, be it because the boundary cannot be recognized without doubt (for instance, tumors in medical

imagery), or because there indeed is no clear boundary (for instance, density distributions like clouds in weather satellite images).

In summary, both vector and raster representation are important for spatial data management, as each of them has its specific strengths and weaknesses; moreover, both representations are independent from each other in the sense that there exists no lossless transformation between them.

References

- [Baum-93] Baumann, P.: Ein konzeptuelles Informationsmodell für Visualisierungsdatenbanken. PhD Thesis, TH Darmstadt, Darmstadt/Germany 1993.
- [Bouk-70] Bouknight, W.; A Procedure for the Generation of Three Dimensional Halftoned Computer Graphics Presentations. Comm. ACM Vol. 13 No. 9, 1970, pp. 527-536.
- [Gour-71] Gouraud, H.; Continuous Shading of Curved Surfaces. IEEE Transactions on Computers, pp. 623-629, June 1971.
- [Kröm-91] Krömker, D.: Visualisierungssysteme - Strukturen, Analysen und Verfahren zur Leistungssteigerung durch einen zum Strukturspeicher erweiterten Bildspeicher. PhD Thesis, TH Darmstadt, 1991.
- [McCo-87] McCormick, B.; DeFanti, T.; Brown, M. (eds.): Visualization in Scientific Computing. ACM Computer Graphics, Vol. 21, No. 6, November 1987.
- [Mort-85] Mortenson, M.: Geometric Modelling. John Wiley and Sons, 1985.
- [Phon-75] Phong, B.; Illumination for computer-generated pictures. Comm ACM, Vol. 18, No. 8, pp 287-296. 1975.