Efficient and Flexible Web Access to Art-Historical Image Collections

Matthias Wagner Stefan Holland Werner Kießling
Institut für Informatik, Universität Augsburg
D–86135 Augsburg, Germany
{wagner,holland,kiessling}@informatik.uni-augsburg.de

ABSTRACT

Modern object-relational database systems are capable of managing multimedia data, e.g. image, video and audio. In this paper we study how such universal database systems can be used for implementing self-tuning and adaptive digital image archives. The presented framework is based on the idea that multimedia objects are often stored redundantly to support broadest system access for diverse clients from heterogenous environments. There is a large variety of ways to store and deliver multimedia data in various formats. Partitioning the data into formats that are physically stored in the database and those that are converted into any delivery format on-demand poses a nontrivial optimization problem. We present a new optimization algorithm which has been implemented and evaluated with the commercial database system DB2 Universal Database from IBM. The work is part of the HERON project, a project committed to the creative use of computer science and technology to advance the study of arts and humanities.

Keywords

Web access, format optimization, multimedia delivery, image databases, applications in the arts and humanities

1. INTRODUCTION

Multimedia digital libraries and the WWW hold great promise for providing wider access to the world's cultural heritage. Museum collections and information in digital form can be distributed and used for many different purposes, from enhancement of scholarship and teaching to personal exploration and enjoyment. To take advantage of these opportunities, however, digital collections need to be useful to a wide audience beyond a single short-term project. Databases for arts and humanities must serve various types of users from extremely heterogeneous environments at different stages of work. Current and anticipated users and uses should be considered when setting up a digital collection. Care must be taken to develop a strategy that minimizes limitations, does not foreclose future options and offers a likely upgrade path.

Developments in Graphical User Interface software for navigating the Internet and the WWW with HTML-browsers or specialized Java-GUIs can offer broad access to multimedia libraries. Providing access to multimedia collections using these different interfaces, however, requires multimedia-storage formats to match the capability of Internet browsers. Especially for large image archives, where system architectures and network topologies become significant concerns, the file formats to be stored and delivered have to be identified carefully [6; 11]. Compromises in one area often affect performance in another. Therefore, trade-offs between storage cost and capacity on one hand and access speed on the other must be examined closely when setting up a networked image database.

Object-relational database systems like the DB2 Universal Database [3] can be deployed to monitor multimedia web sites and image archives. But the image data stored and delivered by such universal database systems is often redundant, because storage formats are heavily interrelated by conversion tools. They may differ in aspects such as compression, color depth and resolution. In this paper we propose a new extension to the storage and delivery of images in todays multimedia databases which enables the dynamic optimization of image delivery. This is done by giving the option either to physically represent a multimedia object or to compute it from others. Given a set of possible conversions between multimedia objects there is generally more than one way of partitioning the formats into physically stored and computed ones. By applying a cost function the database server determines its specific optimal choice considering system dependent aspects such as database access profile, available storage, server load or network bandwidth. This optimization is a complex task which can be automated and performed periodically.

Recently, adaptive media delivery has been proposed in the literature [8; 2; 10]. Whereas [2] studies a highly abstracted formal object-oriented frameworks for the representation and synthesis of media objects and [10] proposes a two-dimensional model for the cross-media adaption of multimedia content, our work presents an efficient implementation of the formal methods studied in [8] on top of the commmercial database system DB2 Universal Database using a new format optimization algorithm. None of the correlated work discuss practical implementation aspects of commer-

SAC'00 March 19-21 Como, Italy
(c) 2000 ACM 1-58113-239-5/00/003>...>$5.00

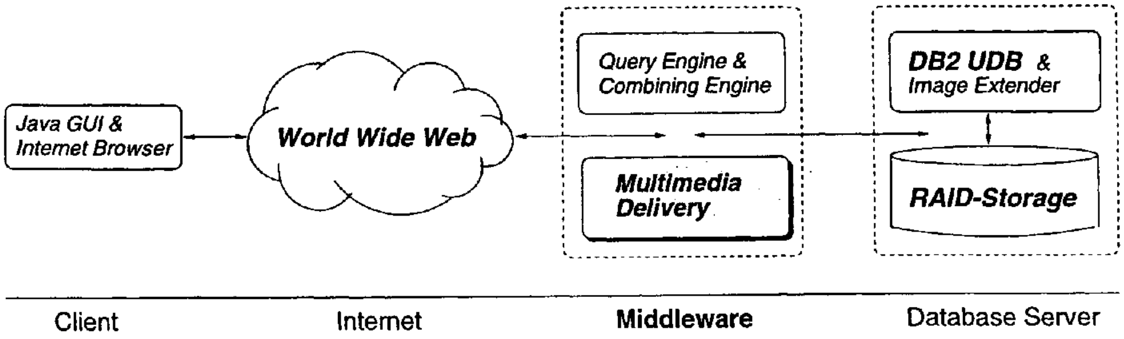

Figure 1: Multimedia delivery within the HERON architecture.

cial database systems. The rest of this paper is organized as follows: An overview of the HERON project and the specific requirements of art-historical image databases is given in section 2. Section 3 shows in brief how image data is managed by the DB2 Universal Database System (UDB) and how the system is used in HERON. Section 4 presents the formal framework for the representation of image formats, conversion constraints and format optimization. This approach is evaluated in section 5. Finally section 6 concludes with a summary and outlines ongoing work.

2. IMAGE DELIVERY FOR HERON

The work presented here is an integral part of the interdisciplinary project HERON (Heraldry Online) [7]. HERON investigates the impact of multimedia applications from the humanities, in particular heraldry [1], on database technology and explores ways to improve networked access to art-historical information. Figure 1 sketches the architecture of the HERON system. Project research areas and middleware aspects of particular concern include the following.

Query Engine: Heraldic research is based on voluminous reference works of images [9]. Since language-bound descriptions and traditional query languages alone seem inappropriate for the retrieval of heraldic images, content-based retrieval techniques based on the visual features of images are applied [5].

Combining Engine: Advanced visual retrieval capabilities typically return a ranked set of image objects as a query result (see figure 2). Since expressing visual similarity will almost always be based on several features simultaneously, e.g. color, texture and shape, an efficient middleware is needed for the combination of multi-feature queries.

Server Scalability: Projects like HERON pose stringent scalability issues, especially for physical data management. E.g., completely digitizing only the heraldic collection of [9] with some 150.000 coat of arms requires about half a terabyte of RAID storage.

Multimedia Delivery: This part addresses the demand of an optimized storage and delivery of multimedia data, in particular images, tailored to the art-historians requirements.

As depicted in figure 1 the HERON query and combining engine are concerned with the processing and the forwarding of requests to the database. Throughout the rest of this paper we will focus on a separate middleware layer for multimedia delivery which provides the optimized delivery of image material to the client.

The way images are used by art historians will determine the specific required resolution and format. HERON takes into account the needs of users from heterogeneous environments at different stages of work. Figure 2 and 3 illustrate some typical user interactions with the system that can in detail be associated with the following activities.

1. Searching&Browsing: Small, low-resolution thumbnail images may be transmitted quickly enabling the remote user to identify a work of art and to browse large query result sets. Further analysis and refinement is often necessary to reduce the size of the initial set retrieved. During this phase images are normally displayed as thumbnails in browseable formats like GIF, JPEG or PNG.

2. Zooming&Analysis: For a close examination of result sets a "medium-resolution" full-screen image may accomplish a full catalog record. Because not all image-file formats can be displayed by all viewers, a large variety of image formats at various levels of quality must be deliverable. Some typical image formats in this stage of work are TIFF, BMP or GIF at diverse resolutions.

3. Storage&Printing: If the retrieved set is to the user's content, a "high-resolution" image must be transmitted for user-side storage and later reference or for printing. Images are often needed at much higher quality for printing than they would be needed for display on a monitor because most printing devices are capable of a much higher resolution than screen displays. In general only lossless formats at very high-level quality or typical print formats are appropriate in this stage of work, including TIFF, PDF or Postscript.

3. IMAGE DELIVERY WITH DB2 UDB

DB2 Universal Database (UDB) is an extensible, object-relational database management system designed to leverage multimedia applications [3]. Like a standard database system UDB stores and manages numeric and character data. In addition, the system enables the storage and retrieval of images in large, complex objects. This extra functionality is bundled in the DB2 Image Extender which defines distinct data types and specific functions for image objects.

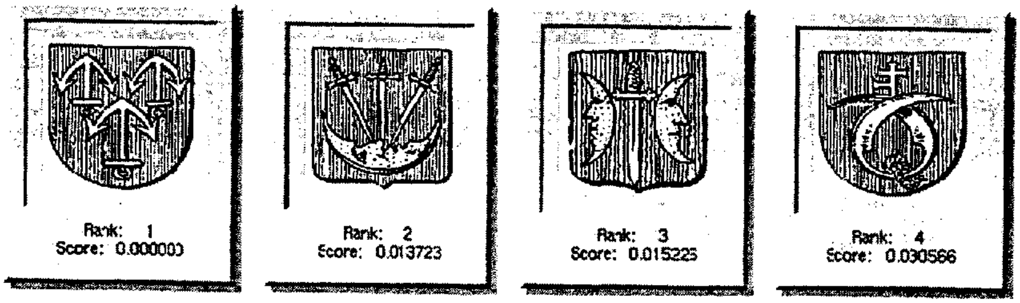

Figure 2: Thumbnailed result set for a HERON image query.

3.1 Reasonable conversion constraints

We have implemented our dynamic optimization of image delivery on top of UDB's Relational Extender for images. Since the implementation is based on standard SQL, JDBC and Java, it can be easily bundled and distributed as an additional DB2 Relational Extender as well as being ported to other universal database systems. The Image Extender itself provides fifteen common image formats for the storage and delivery of image data including the widely used Compuserve GIF format, all TIFF formats, the OS/2 and Microsoft Windows BMP format, the JPEG format or Postscript (PS). Furthermore, the Image Extender allows the scaling of images which are to be stored and delivered to virtually any resolution. We describe formats at certain quality levels relative to the original image scan-size, e.g. TIFF-100 denotes the format of an image at its original scan size (100% scaling), whereas TIFF-25 is the format of images that are scaled down by the factor of four.

Committed to high quality of service not all possible options for image conversion with DB2 Image Extender are equally useful in the HERON system. In particular, care must be taken to avoid lossy format conversions. We consider the following types of image conversions as potentially lossy:

1. The conversion of an image in a low resolution to a high resolution, e.g. converting an image of format GIF-25 to GIF-100.
2. The conversion of an image in a lossy format to an image in a lossless format, e.g. it is not a good idea to derive a 24-bit TIFF-100 image from 8-bit JPEG-100.

Our implementation avoids lossy image conversions by explicitly restricting the possible conversion options of UDB's Image Extender by reasonable conversion constraints.

3.2 The optimization problem

The major requirement of the HERON image archive is to provide image data in a large variety of formats and resolutions. At first glance, one of the following extreme approaches could implement this.

On-demand image conversion: One simple strategy for offering all images of an archive in any format and at any quality level is to store images in a representative format which allows the conversion into all potentially deliverable target formats. In this approach an image which is not physically available in the image base would be converted into any target format at run-time. However, high-resolution images for art-historical purposes can be very large — a 24-bit full-screen image, e.g. in the TIFF format, to be displayed on a workstation can easily occupy 6 MBytes — and the explicit online conversion of derived small versions of these images, like for instance thumbnails, can be computational expensive (we measured about 15 seconds for the conversion of such an image into a GIF thumbnail on a high-end workstation, see section 5). On the other hand, delivering a high-resolution format of 6 MBytes over a low-bandwidth connection may lead to very long delays, e.g. about 10 minutes for a typical ISDN connection.

Off-line image conversion: Physically storing all images at all required quality levels and formats in the database guarantees direct and fast delivery from disk.

But this approach, as it might exceed the available mass storage and financial budget, seems inappropriate either. E.g., for ambitious projects like HERON this would exceed cost-effective use of RAID storage subsystems. In addition, limiting the image database to a certain set of deliverable formats would clearly foreclose future upgrades of the library for new multimedia technology. Thus a practical implementation supporting optimized image delivery must be capable of managing the trade-off between on-demand ("store-one") and off-line ("store-all") image conversion.

Optimized image delivery: The framework presented in this paper is to provide a cost-optimal solution to this time/space trade-off. The tackled optimization problem which is ubiquitous in multimedia database systems is illustrated in figure 4 and can be stated as follows: Given a set of interrelations between the objects of a multimedia database server, what is an optimal partition of the objects formats into those that are physically represented and those that are converted from the physically represented ones?

4. FORMAL OPTIMIZATION MODEL

In this section we describe the algorithm for an optimal partition of formats. The following formal framework for image attributes (formats and resolutions), conversion constraints and format optimization is a refinement of the more general approach from [8] (for details see [12]).

An image format M together with its resolution $R \in \mathcal{R}_M$ is modeled as an attribute $A = (M, R)$, where $\mathcal{R}_M = [R_1, R_2, \ldots, R_n]$ is a list of the supported resolutions for format M

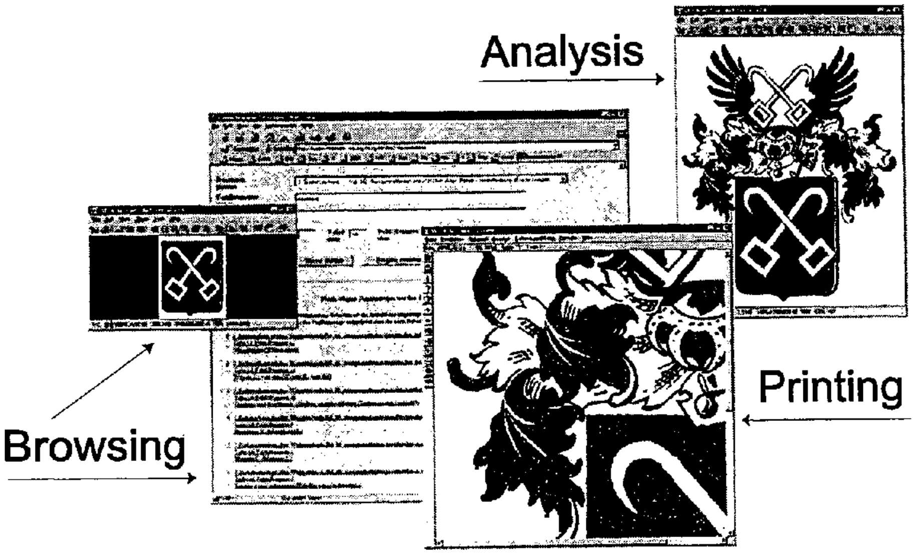

Figure 3: Typical user interaction with the HERON system.

in ascending order. Note that attributes have already been informally denoted as M–R, e.g. TIFF-100 in figure 4. Let $\mathcal{A}$ denote a set of attributes. Format interrelations between attributes $A, A' \in \mathcal{A}$ are modeled as functional constraints, i.e. $fc \equiv (A = f(A'))$. A is called the head attribute of fc and we say that A depends directly on A', denoted as $A \xleftarrow{\mathcal{F}} A'$. Let $\mathcal{F}$ be a set of conversion constraints on formats in $\mathcal{A}$. A functional base $\mathcal{F}_{comp} \subseteq \mathcal{F}$ is a subset partitioning the attributes such that some are physically represented and the rest can uniquely be computed observing conversion constraints. We denote the set of head attributes of $\mathcal{F}_{comp}$ as $\mathcal{A}_{comp}$, i.e. the set of attributes which are computed on-demand. $\mathcal{A}_{phys}$ denotes the remaining attributes which must be physically represented, i.e. $\mathcal{A}_{phys} = \mathcal{A} \setminus \mathcal{A}_{comp}$.

4.1 Optimization algorithm

The optimization problem can now be formally stated as: Determine a functional base optimal w.r.t. a cost function! Basically, the optimal functional base can be determined brute force by evaluating all potential functional bases, but the high complexity of this basic approach seems prohibitive for large database designs. According to [8]: For $n = |\mathcal{A}|$ this basic optimization algorithm has a worst case complexity of $O((n+1)^n \cdot n^2)$. In the following we will focus on a seach space reduction to reduce the complexity of the basic optimization and to speed-up the computation. By reasonable assumptions the search can be restricted to some functional bases which appear to be promising.

4.1.1 Search space reduction

The attribute $A_{min} \in \mathcal{A}$ is named minimal iff for all $A \in \mathcal{A} \setminus \{A_{min}\} : A \xleftarrow{\mathcal{F}} A_{min}$. Note that we referred to a A_{min} in section 3 as a "representative" format. A_{min} is capable of deriving all formats and intended to be an attribute that must be physically stored in the database. Throughout this paper, for any given $\mathcal{A}$ we assume that $A_{min} \in \mathcal{A}$ exists. Furthermore we assume that:

1. A linear order on the attributes associated with a single format is induced by $\mathcal{F}$, i.e. given $(M, R_i), (M, R_j) \in \mathcal{A}$ with $R_i < R_j$ then $(M, R_i) \xleftarrow{\mathcal{F}} (M, R_j)$.
2. For $A, A' \in \mathcal{A}$ such that $A \xleftarrow{\mathcal{F}} A'$, the dependency of A on A' is associated with exactly one functional constraint.

These assumptions are fulfilled by most common image conversion tools and image read/write database extensions. For the DB2 Image Extender the set of possible format conversions can be naturally condensed to fulfill the restrictions. For $\mathcal{A}$ and $\mathcal{F}$ satisfying our assumptions the algorithm restrictedFb determines a restricted functional base:

```
restrictedFb($A_{min}$, $\mathcal{A}_{phys}$, $\mathcal{A}$, $\mathcal{F}$) : $\mathcal{F}_{comp}$
  begin
    $\mathcal{F}_{comp} := \emptyset$;
    foreach $A \in \mathcal{A} \setminus \mathcal{A}_{phys}$ do
      $(A_{pred}, f_{pred})$ := pred($A$, $\mathcal{A}_{phys}$, $\mathcal{F}$);
      if $f_{pred} \neq$ null then
        insert($f_{pred}$, $\mathcal{F}_{comp}$);
      else
        insert($(A = f(A_{min}))$, $\mathcal{F}_{comp}$);
    done
  end

pred($A$, $\mathcal{A}_{phys}$, $\mathcal{F}$) : $(A_{pred}, f_{pred})$
  begin
    $A_{pred} := f_{pred} :=$ null;
    $(M, R) := A$;
    while $R < last(\mathcal{R}_M)$ do
      $R := next(R, \mathcal{R}_M)$;
      if $(M, R) \in \mathcal{A}_{phys}$ then
        $A_{pred} := (M, R)$; $f_{pred} := (A = f(M, R))$;
        break;
    done
  end
```

In the resulting base every format at a certain resolution, which is not directly stored in the database, is preferably derived from the same format physically stored in the next higher resolution, or otherwise from the minimal attribute. For instance, given $\mathcal{A}$, $\mathcal{F}$ in part 1 of figure 4 and A_{min} =

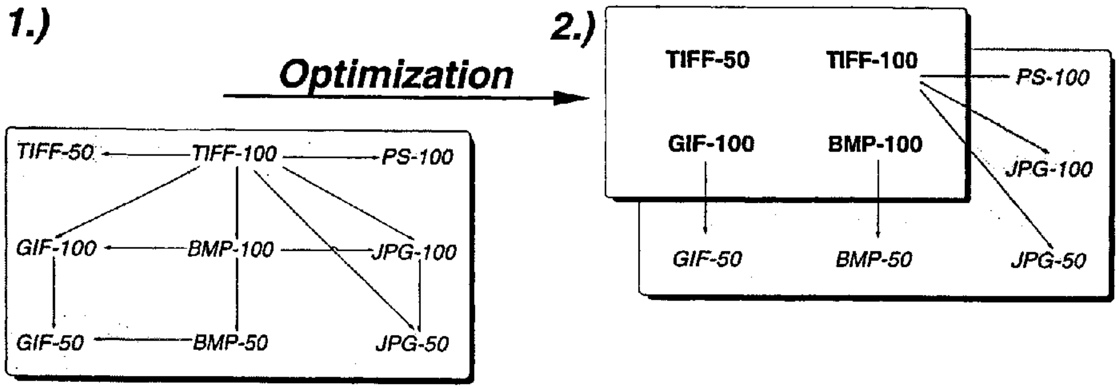

Figure 4: Format optimization for efficient image delivery.

TIFF-100, $\mathcal{A}_{phys}$ = {TIFF-100, TIFF-50, GIF-100, BMP-100} restrictedFb computes the functional base depicted in part 2 of figure 4. The implemented algorithm aims at a fast search but is currently rather restricted, e.g. given a minimal format A_{min} = TIFF-100, A_1 = TIFF-25 and A_2 = GIF-25 with $A_{min}, A_1 \in \mathcal{A}_{phys}$, $A_2 \in \mathcal{A}_{comp}$ then restrictedFb selects A_{min} as the source format to derive A_2 while the conversion to this format could be done with less computational effort from A_1.

Algorithm restrictedFb uses the subroutine pred which, given an attribute A, searches the set $\mathcal{A}_{phys}$ for the format A in the next higher resolution. Moreover, basic functions on lists are assumed: Let l be a list, e a list element. Then $last(l)$ will determine the last element of l, $next(e,l)$ determines the element following e in l and $length(l)$ calculates the length of list l. E.g., let $l = [25, 50, 75, 100]$, then $last(l) = 100$, $next(50, l) = 75$) and $length(l) = 4$.

The optimization algorithm optimalFb can now be easily defined. The algorithm exhaustively searches all possible attribute partitions, sets them in correspondence with their restricted functional base and evaluates them w.r.t. a given cost function to determine the optimal base:

```
optimalFb($A_{min}$, $\mathcal{A}$, $\mathcal{F}$, $cost$): ($\mathcal{F}_{opt}$, $min$)
  begin
      $\mathcal{F}_{opt} := \emptyset$; $min := cost(\emptyset)$;
      foreach $\mathcal{A}_{phys} \subseteq \mathcal{A} \setminus \{A_{min}\}$ do
          $\mathcal{F}_{comp}$ := restrictedFb($A_{min}$, $\mathcal{A}_{phys}$, $\mathcal{A}$, $\mathcal{F}$);
          if $cost(\mathcal{F}_{comp}) \leq min$ then
              $\mathcal{F}_{opt} := \mathcal{F}_{comp}$; $min := cost(\mathcal{F}_{comp})$;
      done
  end
```

4.1.2 *Optimization complexity*

Let $n = |\mathcal{A}|$ and $m = \max\{length(\mathcal{R}_M) \mid M$ an image format$\}$. Then optimalFb has a worst case complexity of $O(2^{n-1} \cdot n \cdot m)$.

This holds because in optimalFb all elements from the powerset of $\mathcal{A} \setminus \{A_{min}\}$ (with size 2^{n-1}) are completed to restricted functional bases. The completion to a restricted base is done by restrictedFb in linear time w.r.t n and m, respectively.

For practical settings of n and m we note substantial savings in the computation of algorithm optimalFb compared to the basic algorithm:

settings for n, m	basic algorithm $(n+1)^n \cdot n^2$	optimalFb $2^{n-1} \cdot n \cdot m$
$n = 10, m = 4$	$\approx 2{,}6 \cdot 10^{12}$	20 480
$n = 9, m = 4$	$\approx 8{,}1 \cdot 10^{10}$	9 216
$n = 8, m = 4$	$\approx 2{,}7 \cdot 10^{9}$	4 096
$n = 7, m = 4$	102 760 448	1 792

4.2 Cost functions

Now we discuss cost functions tailored to the HERON image collection. An appropriate cost function has to consider access costs and storage costs of the database server. We define total costs as

$$\begin{aligned} cost(\mathcal{F}_{comp}) \;\; := \;\; & (1-z) \cdot store\_cost(\mathcal{F}_{comp}) + \\ & z \cdot acc\_cost(\mathcal{F}_{comp}) \; . \end{aligned}$$

By adjusting the parameter $z \in [0, 1]$ the overall behavior of the database server is influenced. For small values of z the optimization will try to minimize the storage costs even if this raises the costs for image access. On the other hand, given sufficient storage space we may decide to optimize image access instead, choosing a z value of almost one.

The functions $store\_cost$ and acc_cost strongly depend on the intended application of the optimization framework. In the HERON system we set $store\_cost$ to the total size of images physically stored in the formats from $\mathcal{A}_{phys}$. The access costs acc_cost are modeled by the sum of the access costs for the particular attribute. The access costs for a single attribute differ for physically represented and computed attributes. In the physical case we assume the costs to be proportional to the average size. For a computed attribute we have to consider the access costs of the attribute it is computed from and the computation costs themselves. Let $A, A' \in \mathcal{A}$ and $total\_acc(A)$ the number of images accessed in format A. Moreover, let $fc \in \mathcal{F}_{comp}$ with $fc \equiv (A = f(A'))$ and $avg\_comp(fc)$ the average computation time for fc. Then the overall access costs are defined as

$$acc\_cost(\mathcal{F}_{comp}) \;\; := \;\; \textstyle\sum_{A \in \mathcal{A}} total\_acc(A) \cdot attr\_acc(A) \; ,$$

$$attr\_acc(A) \;\; := \;\; \begin{cases} c_A \cdot avg\_size(A) \; , \; A \in \mathcal{A}_{phys} \\ avg\_comp(fc) + attr\_acc(A') \; , \\ \qquad\qquad A \in \mathcal{A}_{comp}, A' \in \mathcal{A}_{phys} \; . \end{cases}$$

Choosing the parameter c_A typically depends on the used mass storage devices. Note that the average computation

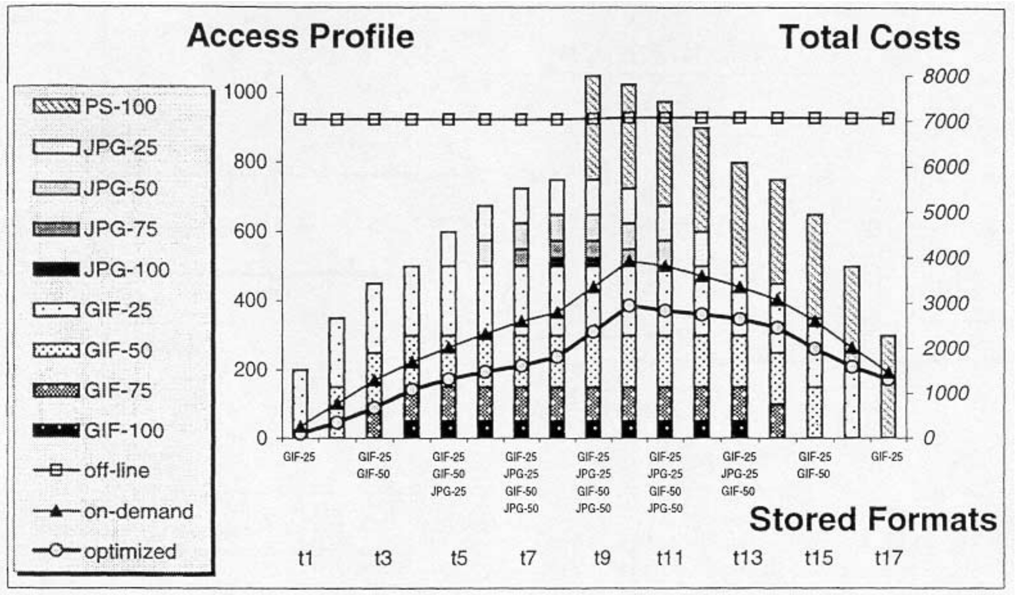

Figure 5: Optimized delivery vs. on-demand and off-line conversion.

costs for a conversion *avg_comp(fc)* are influenced by the workload on the host.

5. SAMPLE OPTIMIZATION RESULTS

In this section we evaluate our framework based on a scenario of varying access profiles at the HERON image server. The testing was performed on an IBM RS/6000 workstation running IBM DB2 Universal Database V5.2 for AIX. The scenario under consideration is motivated by the art-historians stages of work with HERON introduced in section 2: From "searching and browsing" over "zooming and analysis" to "storage and printing". In this example images are accessed in the formats GIF, JPG and PS at resolutions 25, 50, 75 and 100. A heraldic collection of 500 images with an average size of 1 MByte was selected for this scenario. Figure 5 displays the corresponding access profiles together with the arising costs for optimized, off-line and on-demand image delivery. The x-axis of the diagram represents time and is labeled with timestamps, e.g. $t1$. Note that we omit the decorations of all even points in time ($t2$, $t4$ etc.) to keep the illustration compact. In each timestamp the optimization algorithm evaluates all possible database configuration w.r.t. the given cost function (and access profile) and determines the cost-optimal one. Resulting changes in $\mathcal{A}_{phys}$ are depicted as additional decorations of the x-axis, e.g. at $t3$ GIF-25 and GIF-50 are added to the set of physically stored formats. The total costs for this optimized delivery is displayed by the curve "optimized", whereas the curves "off-line" and "on-demand" represent the alternative naive delivery approaches. All curves are associated with the secondary y-axis.

The choice of A_{min} influences the performance of the system and the cost savings of the optimization framework. Throughout this scenario we choose A_{min} = TIFF-100 and this is the only physically stored attribute in $t1$, i.e. $\mathcal{A}_{phys} = \{A_{min}\}$. This is a sensible choice, since images are typically inserted into the heraldic collections of HERON as high quality scans. To clarify the impact of A_{min} on the optimization result we re-performed the same scenario with varying choices for A_{min}. The following table summarizes the resulting average savings in costs of the optimized delivery compared to the standard approaches.

A_{min}	savings to on-demand	savings to off-line
TIFF-100	32%	75%
BMP-100	37%	68%
JPG-100	36%	71%

Depending on the cost parameter z higher expenses for storage forced by the optimization on the one hand are leading to access efficiency gains on the other. In this scenario we claim that the ratio of storage to access is 1/500 and thus set z to the value $2 \cdot 10^{-3}$. In addition figure 6 abstracts from the cost parameter z by splitting the curves "on-demand" and "optimized" from above into single diagrams for storage and access costs which do not depend on z. As with z, the parameter c_A of the cost function *attr_acc* is subject to the specific system context of the optimization framework. In this sample scenario we assumed the average throughput of the database server disks to be 20 MByte/s = 20000 Byte/ms. With milliseconds (ms) as the unit of measurement for time we set c_A to $\frac{1}{20000} = 5 \cdot 10^{-5}$.

6. SUMMARY AND OUTLOOK

In this paper we discussed an optimization problem that is ubiquitous in digital image collections and web-enabled multimedia databases. A variety of different image formats required by the WWW clients is accompanied by many conversion tools interrelating those formats. Thus the server can compute some formats as well as physically represent them. We formalized the conversion tools as functional constraints among formal attributes and presented an algorithm to determine an optimal division of physically represented and computed attributes. The work presented here is concerned

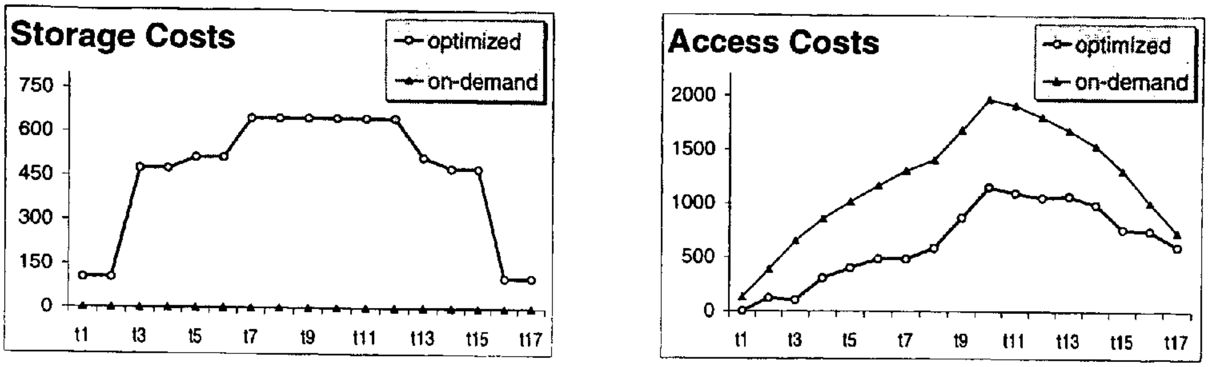

Figure 6: Varying storage and access costs.

with the efficient delivery of image material, e.g. query results, to Web clients. The selection of an image format can also have an impact on the performance and quality of the content-based retrieval techniques applied in HERON, however this has not been exploited yet. A first evaluation with the HERON image collection has shown substantial savings compared to off-line and on-demand conversion of images.
Modern database technologies, like the presented work, must be employed to bring the full benefit of digital libraries to arts and humanities. However, the established optimization framework is general enough to be applied in other domains beyond this field. In virtually any application domain where large-scale universal database systems are applied for the efficient and flexible delivery of multimedia data, self-organization and self-tuning of these systems is mandatory [4]. With large-scale database systems, minimizing the database administration effort for performance tuning becomes important. Intuitive, graphical database administration tools can ease database tuning, scaling and optimization. Therefore, in addition to the core optimization framework a prototypical dba-tool with a Java-GUI has been implemented and will be enhanced in the future.
With the advent of new Internet standards, like the extensible markup language XML and the corresponding stylesheet language XSL, content and display of WWW pages will conceptually be separated. The presented framework allows to publish content once and to adapt content display to forthcoming standards. We are already considering advanced value-added services for the dissemination of images, e.g. the integrated delivery of web and facsimile formats, and intend to take applications from mobile and wireless environments into consideration.

Acknowledgements

We are grateful to W.-T. Balke, T. Ehm and T. Birke for helpful comments and suggestions. The HERON-project is funded by the Deutsche Forschungsgemeinschaft DFG.

7. REFERENCES

[1] C. Boutell. *Heraldry.* Frederick Warne & Co. Ltd., London UK, rev. ed. by j. p. brooke-little edition, 1958.

[2] K. S. Candan, P. V. Rangan, and V. S. Subrahmanian. Collaborative Multimedia Systems: Synthesis of Media Objects. *IEEE Trans. Knowledge and Data Engineering*, 10(3):433–457, May/June 1998.

[3] D. Chamberlin. *A Complete Guide to DB2 Universal Database.* M. Kaufmann, 1998.

[4] S. Chaudhuri and V. R. Narasayya. AutoAdmin 'What-if' Index Analysis Utility. In L. M. Haas and A. Tiwary, editors, *Proc. of ACM SIGMOD Conference*, pages 367–378, Seattle, 1998.

[5] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by image and video content: The QBIC system. *Computer*, 28(9):23–32, Sept. 1995.

[6] B. Furth, S. W. Smoliar, and H. Zhang. *Video and Image Processing in Multimedia Systems.* Kluver Academic Publishers, Boston, 1995.

[7] W. Kießling, K. Erber-Urch, W.-T. Balke, T. Birke, and M. Wagner. The HERON Project — Multimedia Database Support for History and Human Sciences. In J. Dassow and R. Kruse, editors, *Proc. of the GI Annual Conference INFORMATIK'98*, volume XI of *Informatik Aktuell*, pages 309–318. Springer, Sept 1998.

[8] G. Köstler, W. Kowarschick, and W. Kießling. Client-Server Optimization for Multimedia Document Exchange. In *Proc. of the Fifth International Conference on Database Systems for Advanced Applications*, pages 135–144, Melbourne, Australia, April 1997.

[9] J. Siebmacher. *Siebmachers großes Wappenbuch.* Bauer & Raspe, Neustadt a.d. Aisch, reprint edition, 1856.

[10] J. R. Smith, R. Mohan, and C.-S. Li. Scalable Multimedia Delivery for Pervasive Computing. In *Proc. of the 7th ACM International Multimedia Conference*, pages 131-140, Orlando, Florida, 1999.

[11] V. S. Subrahmanian. *Principles of Multimedia Database Systems.* Morgan Kaufmann, San Francisco, 1998.

[12] M. Wagner, S. Holland, and W. Kießling. Efficient and Flexible Multimedia Delivery with Universal Database Systems. Technical Report 1999-02, Universität Augsburg, August 1999.