OPTIMAL ALLOCATION OF TWO-DIMENSIONAL DATA
(Extended Abstract)

Khaled A. S. Abdel-Ghaffar*[1] and Amr El Abbadi**[2]

[1] Department of Electrical and Computer Engineering, University of California, Davis, CA 95616, USA
[2] Department of Computer Science, University of California, Santa Barbara, CA 93106, USA

Abstract. Efficient browsing and retrieval of geographically referenced information requires the allocation of data on different storage devices for concurrent retrieval. By dividing a two dimensional space into tiles, a system can allow users to specify regions of interest using a query rectangle and then retrieving all information related to tiles overlapping with the query. In this paper, we derive the necessary and sufficient conditions for strictly optimal allocations of two-dimensional data. These methods, when they exist, guarantee that for any query, the minimum number of tiles are assigned the same storage device, and hence ensures maximal retrieval concurrency.

1 Introduction

Information displayed on a two dimensional screen is often used as a tool for supporting the retrieval of more detailed data that is associated with a specific part of the screen. A common example in geographic applications is to display a schematic or low resolution map of the world on a screen. A user may then specify a region of the map (via a two dimensional rectangle) and request the displaying of a higher resolution satellite image of that region. Alternatively, the user may request that pictures of famous historical monuments located in cities in the selected region be displayed. Efficient support of such queries is quite important for image databases in particular, and for browsing geographically referenced information in general. In the Alexandria Digital Library project [Smi96], a large satellite image is often divided into tiles and each tile is decomposed using wavelet decomposition [VH92]. A wavelet decomposition of an image results in a lower resolution image of the original image, in addition to higher order coefficients that can be used to retrieve higher resolution versions of the same image. Similar approaches are common in other systems for browsing large image

* This author was supported in part by the National Science Foundation under grant NCR-9115423.
** This author was supported in part by the National Science Foundation under grant IRI-9411330.

databases [FBF+94,OS95]. A user would usually browse the lower resolution images fast and then specify regions to be expanded to display at a higher resolution. Such reconstruction requires the retrieval of the higher resolution coefficients (or, in general, components) for the various tiles that overlap with the specified region.

In this paper we address the problem of efficiently retrieving data that is associated with a rectilinearly oriented rectangle defined on a two dimensional grid. Since the data associated with individual tiles of the grid can be quite large, we would like to store them on parallel I/O devices in such a way that for a given query, retrieval from these parallel devices can occur concurrently. Ideally, the information related to each tile would be stored on a distinct I/O device and for any query all tile information would be retrieved concurrently. However, in general, the number of I/O devices is much less than the number of tiles. In this paper, we first define an optimality criterion for the allocation of two dimensional data and then provide both necessary and sufficient conditions for optimal allocation methods. In the cases where an optimal allocation method exists, we provide simple allocation methods that can be easily implemented. Previous research concentrated on efficient retrieval of images for circular queries [CR93] and, more generally, on the disk allocation of relational databases for range queries [Fal88,LSR92,CC92], as well as the organization of raster-graphics memory [CLRS86]. We discuss their relationship to our work as well as future extensions in the conclusion of the paper.

2 Problem Statement

Consider the domain of discourse, an $n_1 \times n_2$ array whose elements (i, j), where $0 \leq i \leq n_1 - 1$ and $0 \leq j \leq n_2 - 1$, are called *tiles*. Each tile can be viewed as containing more detailed information about the area it covers. For example, if the array is a low resolution image of a geographic region, the higher resolution wavelet coefficients may be associated with the individual tiles. Alternatively, pictures of famous monuments may be associated with the tiles where they are geographically located. Given two tiles (i_1, j_1) and (i_2, j_2), where $i_1 \leq i_2$ and $j_1 \leq j_2$, we define the two dimensional query

$$\mathrm{R}[(i_1, j_1), (i_2, j_2)] = \{(i, j) : i_1 \leq i \leq i_2 \text{ and } j_1 \leq j \leq j_2\}.$$

This represents the $(i_2 - i_1 + 1) \times (j_2 - j_1 + 1)$ rectangle with (i_1, j_1) and (i_2, j_2) as opposite corner tiles. The area of this rectangle is $(i_2 - i_1 + 1)(j_2 - j_1 + 1)$, which equals the number of tiles in the rectangle.

Each tile (i, j) in the $n_1 \times n_2$ array is assigned a number $f(i, j)$ in the set $\{0, 1, \ldots, M - 1\}$. For efficient retrieval, this set of numbers refer to M I/O storage devices on which the tile-related information can be stored. We say that f is an M-assignment for the $n_1 \times n_2$ array. Clearly, any rectangle of area A has at least $\lceil A/M \rceil$ tiles that are assigned the same number. We say that the M-assignment f is *strictly optimal* if no rectangle of area A has more than $\lceil A/M \rceil$

tiles that are assigned the same number. In particular, in a strictly optimal M-assignment, the numbers assigned to the tiles in any rectangle of area at most M are distinct. Hence, given a grid with a strictly optimal M-assignment, all tiles of any rectangle query of size M will reside on different I/O devices and hence can be retrieved in parallel, ensuring minimal retrieval time.

We may identify the assignment f by an $n_1 \times n_2$ array where entry (i, j) is given by $f(i, j)$. For example, the 5-assignment

0	1	2	3	4	0	1	2
2	3	4	0	1	2	3	4
4	0	1	2	3	4	0	1
1	2	3	4	0	1	2	3
3	4	0	1	2	3	4	0
0	1	2	3	4	0	1	2

is strictly optimal for the 6×8 array, while the 5-assignment shown below is not.

0	1	2	0	1	2	0	1
3	4	0	3	4	0	3	4
0	1	2	0	1	2	0	1
3	4	0	3	4	0	3	4
0	1	2	0	1	2	0	1
3	4	0	3	4	0	3	4

Theorem 1 is the main result of this paper. It gives necessary and sufficient conditions for the existence of strictly optimal M-assignments for $n_1 \times n_2$ arrays in terms of M, n_1, and n_2.

Theorem 1. *A strictly optimal M-assignment exists for an $n_1 \times n_2$ array if and only if any of the following conditions holds:*

- $\min\{n_1, n_2\} \leq 2$,
- $M \in \{1, 2, 3, 5\}$,
- $M \geq n_1 n_2 - 2$,
- $M = n_1 n_2 - 4$ *and* $\min\{n_1, n_2\} = 3$,
- $M = 8$ *and* $n_1 = n_2 = 4$.

The proof of this result is presented in the following two sections. In the next section, we describe strictly optimal M-assignments if any of the conditions stated in Theorem 1 is satisfied. In Section 4, we complete the proof of the theorem by showing that no strictly optimal assignments exist for other choices of M, n_1, and n_2.

3 Strictly Optimal Assignments

We start by establishing strictly optimal assignments when one of the two dimensions of the array is less than or equal to two or when M is restricted to one, two, three, or five.

Lemma 2. *Suppose that $n_1 \le 2$ or $M \in \{1,2,3,5\}$. Consider the assignment f given by $f(i,j) = \lfloor M/2 \rfloor i + j \bmod M$. Then, any rectangle of area A has no more than $\lceil A/M \rceil$ tiles that are assigned the same number. In particular, the assignment f is strictly optimal.*

Proof. We consider different cases.

- $n_1 = 1$: In this case, the tiles in the array are $(0,0), (0,1), \ldots, (0, n_2 - 1)$ and any rectangle of area A is a $1 \times A$ rectangle. Since $f(i,j) = j \bmod M$, and the numbers $t, t+1, \ldots, t+M-1$ are distinct modulo M, for any integer t, it follows that the lemma holds.
- $n_1 = 2$: In this case, any rectangle of area A is either a $1 \times A$ or $2 \times \lfloor A/2 \rfloor$ rectangle. Using the same argument used in case $n_1 = 1$, it can be shown that the lemma holds for $1 \times A$ rectangles. If M is even, then the numbers $(M/2)i + j$, where $0 \le i \le 1$ and $t \le j \le t + M/2 - 1$, are distinct modulo M for any integer t. Hence the lemma holds for $2 \times \lfloor A/2 \rfloor$ rectangles. If M is odd, then the numbers $((M-1)/2)i + j$, where $0 \le i \le 1$ and $t \le j \le t + M - 1$, assume each value modulo M exactly twice for any integer t. Hence, it suffices to consider $2 \times \lfloor A/2 \rfloor$ rectangles where $\lfloor A/2 \rfloor \le M - 1$. On the other hand, the numbers $((M-1)/2)i + j$, where $0 \le i \le 1$ and $t \le j \le t + (M-3)/2$, are distinct modulo M for any integer t. Hence, we may assume that $(M+1)/2 \le \lfloor A/2 \rfloor \le M - 1$. From the above, it follows that the rectangle $\mathrm{R}[(0,t),(1,t+\lfloor A/2 \rfloor - 1)]$ has at most one tile (i', j'), where $t \le j' \le t + (M-3)/2$, and at most one tile (i'', j''), where $t + (M-1)/2 \le j'' \le t + \lfloor A/2 \rfloor - 1 \le t + M - 2$, that are assigned the same number. Hence, there are at most two distinct tiles that are assigned the same number in any $2 \times \lfloor A/2 \rfloor$ rectangle, where $\lfloor A/2 \rfloor \ge (M+1)/2$. This proves the lemma.
- $M = 1$: In this case, all tiles are assigned the same number and the lemma holds trivially.
- $M \in \{2,3,5\}$: In this case M is a prime and the M numbers $\lfloor M/2 \rfloor i + j$, where i is fixed and j runs over M consecutive integers, or vice versa, are distinct modulo M. Hence, it suffices to consider $a_1 \times a_2$ rectangles, where $a_1, a_2 \le M - 1$. If $a_1 \le 2$ or $a_2 \le 2$, then the argument used in case $n_1 = 2$ shows that the lemma holds for such rectangles. Therefore, the lemma holds if $M \in \{2,3\}$. If $M = 5$, it is easy to check that any $a_1 \times a_2$ rectangle where $3 \le a_1, a_2 \le 4$ has at most $\lceil a_1 a_2 / 5 \rceil$ tiles assigned the same number. This proves the lemma. □

Next, we establish strictly optimal assignments for the remaining cases listed in Theorem 1.

Lemma 3. *Consider the following M-assignments f for an $n_1 \times n_2$ array depending on the values of n_1, n_2, and M, where $n_2 \ge n_1 \ge 3$, $0 \le i \le n_1 - 1$, and $0 \le j \le n_2 - 1$:*

- *$M \ge n_1 n_2$: $f(i,j) = n_2 i + j$.*

- $M = n_1n_2 - 1$: $f(i,j) = \begin{cases} 0 & \text{if } (i,j) = (n_1-1, n_2-1), \\ n_2 i + j & \text{otherwise.} \end{cases}$
- $M = n_1n_2 - 2$:

$$f(i,j) = \begin{cases} 0 & \text{if } (i,j) = (n_1-1, n_2-1), \\ n_2 - 1 & \text{if } (i,j) = (n_1-1, 0), \\ n_2(n_1-1)+j-1 & \text{if } i = n_1 - 1 \text{ and } j = 1, \ldots, n_2-2, \\ n_2 i + j & \text{if } i = 0, \ldots, n_1 - 2. \end{cases}$$

- $M = n_1n_2 - 4$ *and* $n_2 > n_1 = 3$:

$$f(i,j) = \begin{cases} 0 & \text{if } (i,j) = (n_1-1, n_2-2), \\ 1 & \text{if } (i,j) = (n_1-1, n_2-1), \\ n_2 - 2 & \text{if } (i,j) = (n_1-1, 0), \\ n_2 - 1 & \text{if } (i,j) = (n_1-1, 1), \\ n_2(n_1-1)+j-2 & \text{if } i = n_1 - 1 \text{ and } j = 2, \ldots, n_2-3, \\ n_2 i + j & \text{if } i = 0, \ldots, n_1 - 2. \end{cases}$$

- $M = 8$ *and* $n_1 = n_2 = 4$: $f = \begin{array}{|c|c|c|c|} \hline 0 & 1 & 2 & 3 \\ \hline 4 & 5 & 6 & 7 \\ \hline 2 & 3 & 0 & 1 \\ \hline 6 & 7 & 4 & 5 \\ \hline \end{array}$.

Then, any rectangle of area A has no more than $\lceil A/M \rceil$ tiles that are assigned the same number. In particular, the assignment f is strictly optimal.

Proof. In each case, we list all pairs of tiles that are assigned the same number.

- $M \geq n_1n_2$: none.
- $M = n_1n_2 - 1$: $((0,0),(n_1-1,n_2-1))$.
- $M = n_1n_2 - 2$: $((0,0),(n_1-1,n_2-1))$, $((0,n_2-1),(n_1-1,0))$.
- $M = n_1n_2 - 4$ and $n_2 > n_1 = 3$: $((0,0),(n_1-1,n_2-2))$, $((0,1),(n_1-1,n_2-1))$, $((0,n_2-2),(n_1-1,0))$, $((0,n_2-1),(n_1-1,1))$.
- $M = 8$, $n_1 = n_2 = 4$: $((0,0),(2,2))$, $((0,1),(2,3))$, $((0,2),(2,0))$, $((0,3),(2,1))$, $((1,0),(3,2))$, $((1,1),(3,3))$, $((1,2),(3,0))$, $((1,3),(3,1))$.

It is easy to check that no rectangle of area at most M contains any of the above pairs. Furthermore, since no three tiles are assigned the same number, the lemma holds. □

4 Necessary Conditions for Strict Optimality

In this section, we conclude the proof of Theorem 1 by showing that no strictly optimal assignments exist if none of the conditions stated in the theorem is satisfied.

Lemma 4. *Consider an $n_1 \times n_2$ array where $n_2 \geq n_1 \geq 3$. If $M \leq 2n_2 - 1$ and $M \notin \{1, 2, 3, 5\}$, then there is no strictly optimal M-assignment for this array.*

Proof. Suppose that f is a strictly optimal assignment. Then, the tiles in the rectangle $\mathrm{R}[(0,0),(1,\lfloor M/2\rfloor-1)]$ are assigned distinct numbers. We may assume that $f(i,j)=\lfloor M/2\rfloor i+j$, where $0\le i\le 1$ and $0\le j\le\lfloor M/2\rfloor-1$. Then, the numbers $0,1,\ldots,2\lfloor M/2\rfloor-1$ are assigned to distinct tiles in the rectangle. The tiles in each of the three rectangles $\mathrm{R}[(0,0),(0,\lfloor M/2\rfloor)]$, $\mathrm{R}[(1,0),(1,\lfloor M/2\rfloor)]$, and $\mathrm{R}[(0,1),(1,\lfloor M/2\rfloor)]$ are assigned distinct numbers. If M is even, this implies that $f(0,0)=f(1,M/2)$ and $f(1,0)=f(0,M/2)$. A similar argument yields $f(2,0)=f(1,M/2)$. Hence, $f(0,0)=f(2,0)$. Since the tiles $(0,0)$ and $(2,0)$ are in a rectangle of area equal to 3 and $M\ge 4$, we get a contradiction. On the other hand, if M is odd, then $f(1,(M-1)/2)\in\{f(0,0),M-1\}$ and $f(0,(M-1)/2)\in\{f(1,0),M-1\}$. Clearly, $f(0,(M-1)/2)\ne f(1,(M-1)/2)$ since the tiles $(0,(M-1)/2)$ and $(1,(M-1)/2)$ are in a rectangle of area equal to 2. Therefore, we may assume, without loss of generality, that $f(1,(M-1)/2)=f(0,0)=0$. The tiles in each of the rectangles $\mathrm{R}[(1,1),(2,(M-1)/2)]$ and $\mathrm{R}[(0,0),(2,0)]$ are assigned distinct numbers. Hence, for $0\le a_2\le(M-3)/2$, $f(2,a_2)\ne 0$, and $f(2,a_2)\ne f(1,a_2')$, where $a_2'=0,1,\ldots(M-3)/2$. In particular, the $(M-1)/2$ tiles $(2,a_2)$, where $a_2=0,1,\ldots(M-3)/2$, are assigned distinct numbers in the set $\{1,\ldots,(M-3)/2,M-1\}$ of $(M-1)/2$ numbers. Since $M\ge 7$ (as M is odd and M not in $\{1,2,3,5\}$), each of the rectangles $\mathrm{R}[(0,0),(2,\lfloor(M-1)/4\rfloor)]$ and $\mathrm{R}[(0,\lfloor(M-1)/4\rfloor),(2,(M-3)/2)]$ has an area of at most M. Hence, for $a_2=0,1,\ldots,(M-3)/2$, $f(2,a_2)\ne f(0,\lfloor(M-1)/4\rfloor)=\lfloor(M-1)/4\rfloor$. This implies that the a_2's cannot be assigned distinct values in the set $\{1,\ldots,(M-3)/2,M-1\}$. This contradiction shows that no strictly optimal assignment exists. □

Lemma 5. *Suppose that an $n_1\times n_2$ array has a strictly optimal M-assignment. Then, for any integer $t\le n_1$,*

$$\sum_{k=\lfloor M/n_2\rfloor+1}^{t}(2\lfloor M/k\rfloor-n_2)\le M-n_2\lfloor M/n_2\rfloor.$$

Proof. Clearly, the lemma holds in case $\lfloor M/n_2\rfloor=0$, i.e., $n_2\ge M+1$. So, assume in the following that $\lfloor M/n_2\rfloor\ge 1$. The $n_2\lfloor M/n_2\rfloor$ tiles (i,j), where $0\le i\le\lfloor M/n_2\rfloor-1$ and $0\le j\le n_2-1$, are assigned different numbers since they are contained in a rectangle of area at most M. Similarly, if $\lfloor M/n_2\rfloor+1\le k\le n_1$, then the tiles (i,j), where $0\le i\le k-1$ and $0\le j\le\lfloor M/k\rfloor-1$, are assigned different numbers and the tiles (i,j), where $0\le i\le k-1$ and $n_2-\lfloor M/k\rfloor\le j\le n_2-1$, are also assigned different numbers. Hence, the $\max\{0,2\lfloor M/k\rfloor-n_2\}$ tiles $(k-1,j)$, where $n_2-\lfloor M/k\rfloor\le j\le\lfloor M/k\rfloor-1$, are assigned distinct numbers that differ from the numbers assigned to the tiles (i,j) where $i\le k-2$. Therefore, for any integer $t\le n_1$, the tiles $(k-1,j)$, where $\lfloor M/n_2\rfloor+1\le k\le t$ and $n_2-\lfloor M/k\rfloor\le j\le\lfloor M/k\rfloor-1$, are assigned distinct numbers that differ from the $n_2\lfloor M/n_2\rfloor$ numbers assigned to the tiles (i,j), where $0\le i\le\lfloor M/n_2\rfloor-1$ and $0\le j\le n_2-1$. Hence,

$$\sum_{k=\lfloor M/n_2\rfloor+1}^{t}(2\lfloor M/k\rfloor-n_2)\le\sum_{k=\lfloor M/n_2\rfloor+1}^{t}\max\{0,2\lfloor M/k\rfloor-n_2\}\le M-n_2\lfloor M/n_2\rfloor.$$

$\square$

Using the above lemmas, we can complete the proof of Theorem 1.

Proof of Theorem 1.

Without loss of generality, we may assume that $n_1 \leq n_2$. If any of the conditions is satisfied, then the result follows from Lemmas 2 and 3. So, assume in the following that none of the conditions holds and, to obtain a contradiction, that a strictly optimal M-assignment exists. If $n_2 = 3$, then $n_1 n_2 = 9$. Since $M < n_1 n_2 - 2$, we have $M = 4$ or 6. By Lemma 4, $M = 6$. However, it is easy to check that there is no strictly optimal 6-assignment for a 3×3 array. Hence, $n_2 \geq 4$. Lemma 4 implies that $M \geq 2n_2$, i.e., $\lfloor M/n_2 \rfloor \geq 2$. Since $M < n_1 n_2$, $\lfloor M/n_2 \rfloor \leq n_1 - 1$. Taking $t = \lfloor M/n_2 \rfloor + 1$ in Lemma 5, we obtain

$$n_2(\lfloor M/n_2 \rfloor + 1) - M \leq 2 \left\lceil \frac{n_2(\lfloor M/n_2 \rfloor + 1) - M}{\lfloor M/n_2 \rfloor + 1} \right\rceil .$$

This implies one of the following conditions:

1. $n_2(\lfloor M/n_2 \rfloor + 1) - M = 1$.
2. $n_2(\lfloor M/n_2 \rfloor + 1) - M = 2$.
3. $n_2(\lfloor M/n_2 \rfloor + 1) - M = 4$ where $\lfloor M/n_2 \rfloor = 2$.

If $\lfloor M/n_2 \rfloor = n_1 - 1$, then each of these conditions implies a condition stated in the theorem. Hence, $\lfloor M/n_2 \rfloor \leq n_1 - 2$. Taking $t = \lfloor M/n_2 \rfloor + 2$ in Lemma 5, we obtain the following for conditions 1, 2, and 3, respectively.

- Condition 1: $n_2(\lfloor M/n_2 \rfloor + 1) - M = 1$ gives $n_2 \leq 2\lceil (n_2+1)/(\lfloor M/n_2 \rfloor + 2) \rceil + 1$. Since $n_2 \geq 4$ and $\lfloor M/n_2 \rfloor \geq 2$, it follows that either $n_2 = 4$ and $\lfloor M/n_2 \rfloor = 2$ or $n_2 = 5$ and $2 \leq \lfloor M/n_2 \rfloor \leq 3$. If $n_2 = 4$ and $\lfloor M/n_2 \rfloor = 2$, then $M = 11$ and $4 = \lfloor M/n_2 \rfloor + 2 \leq n_1 \leq n_2 = 4$, i.e., $n_1 = 4$. However, it is easy to check that no strictly optimal 11-assignment exists for a 4×4 array. If $n_2 = 5$ and $\lfloor M/n_2 \rfloor = 2$, then $M = 14$ and $n_1 = 4$ or 5. Again, it is easy to check that no strictly optimal 14-assignment exits for 4×5 or 5×5 arrays. If $n_2 = 5$, and $\lfloor M/n_2 \rfloor = 3$, then $M = 19$ and $n_1 = 5$. Again, no strictly optimal 19-assignment exits for 5×5 arrays.
- Condition 2: $n_2(\lfloor M/n_2 \rfloor + 1) - M = 2$ gives $n_2 \leq 2\lceil (n_2+2)/(\lfloor M/n_2 \rfloor + 2) \rceil$. Since $n_2 \geq 4$ and $\lfloor M/n_2 \rfloor \geq 2$, it follows that $n_2 = 4$ and $2 \leq \lfloor M/n_2 \rfloor \leq 3$. If $\lfloor M/n_2 \rfloor = 2$, then $M = 10$ and $n_1 = 4$. However, it is easy to check that no strictly optimal 10-assignment exists for a 4×4 array. If $\lfloor M/n_2 \rfloor = 3$, then $n_1 \geq 5$, which contradicts the assumption $n_1 \leq n_2$.
- Condition 3: $n_2(\lfloor M/n_2 \rfloor + 1) - M = 4$ where $\lfloor M/n_2 \rfloor = 2$ gives $n_2 \leq 2\lceil n_2/4 \rceil + 2$, i.e., $n_2 = 4$, 5, or 6. If $n_2 = 4$, then $M = 8$ and $n_1 = 4$, which implies a condition stated in the theorem. If $n_2 = 5$, then $M = 11$ and $n_1 = 4$ or 5. However, it is easy to check that no strictly optimal 11-assignment exists for 4×5 or 5×5 arrays. If $n_2 = 6$, then $M = 14$ and $n_1 = 4$, 5, or 6. Again, no strictly optimal 14-assignment exits for 4×6, 5×6, or 6×6 arrays. $\square$

5 Discussion

In this paper, we derived the necessary and sufficient conditions for the existence of strictly optimal M-assignments of two-dimensional data. We show that except for a few cases, solutions cannot be strictly optimal. However, the results are important in that they have several ramifications from the practical point of view, as well as implications on future research. In terms of allocating tiled images on multiple I/O storage devices, a direct implication of Theorem 1 is that unless the system under consideration has the exact number of I/O devices specified in Theorem 1, the best tiling of a two dimensional grid is one that divides the space into strips (n_1 or n_2 =1) or first into stripes and then each stripe is divided in two (n_1 or $n_2 = 2$).

Alternatively, Theorem 1 states that strict optimality can be achieved if the number of parallel devices is $1, 2, 3$ or 5. Although this may seem quite limited, in fact, for some applications such as wavelet decomposition that require recursive decompositions, multiples of these numbers of storage devices would also result in optimal allocation methods. In particular, each tile in the lower resolution image grid can be expanded into the next higher resolution image using three coefficients [VH92]. Hence, assuming we have 15 parallel I/O devices, we cluster them into 5 groups, each with 3 disks. Theorem 1 is first used to partition the tiles among the 5 groups (there is a strictly optimal solution for this case). Subsequently, within each group, the 3 coefficients are distributed to 3 different devices. In fact, if r levels of decomposition are used, as many as $m * 3^r$, where $m = 1, 2, 3, 5$, I/O devices can be used for any tiling of an image.

A closely related problem to the M-assignment of two dimensional data arises in graphical interfaces where clustered pixels are stored on different memory chips for simultaneous access. Chor, Leiserson, Rivest, and Shearer [CLRS86] present an almost optimal technique based on a Fibonacci lattice to accomplish this task. However, their approach is different in that the memory organization guarantees that given a fixed number M of memory chips, if any rectangle contains less than $M/\sqrt{5}$ pixels (or tiles in our context), then all pixels will reside on different memory chips. In our problem, we are concerned with strict optimality for any rectangle, and not ones whose sizes are a function of the number of memory chips (or in general I/O devices).

In general, the problem of allocating a two dimensional grid on multiple I/O devices can be viewed as an important instance of range queries for multi-dimensional files. Two types of queries have been considered: partial match queries and range queries [AU79,Du86,Sun87,KP88,FM91,AE93]. The problem discussed in this paper can be viewed as a special case of range queries that are limited to two attributes on a relation with n attributes (the two attributes are the x and y axes, and the rest of the attributes correspond to the data associated with each unique combination of x and y). Optimality results for partial match queries have been derived [AE93]. However, for range queries, only a few allocation methods have been proposed [Fal88,GD90,GDQ92,CC92,LSR92,CR93]. Ghandeharizadeh, DeWitt and Qureshi [GD90,GDQ92] demonstrated, using simulation, that declustering data on multiple disks results in significant improvements in response time for range queries. In [Fal88], Faloutsos uses gray code techniques and in [FB93], Faloutsos and Bhagwat use fractals for declustering multi-dimensional files for range queries. Experiments show that declustering using these techniques provide further improvements in response time. The CMD method proposed by Li *et al.* [LSR92] also uses a modulo approach for allocating data on disks, however no optimality conditions have been derived. In [CR93], a declustering algorithm is proposed for images and maps where queries are defined by a circle around a point of interest rather than a rectangle. For a given number of storage devices, the algorithm searches for an allocation method that maximizes the query radius such that no two tiles are on the same device. However, the search is restricted to a certain class of allocation methods called vector based declustering and no closed form solution is presented. Chen and Chang in [CC92] study the disk modulo allocation method for range queries and give sufficient (but not necessary) conditions for strict optimality. This method was originally developed by Du and Sobolewski for partial match queries [DS82]. For two-dimensions, and using the language of our paper, the allocation method assigns the number $x + y \bmod M$ to the tile represented by (x, y). It is shown in [CC92] that the method is strictly optimal if $M = 2$ or 3. This result, in two-dimensions, follows from Lemma 1 which states that the assignment $\lfloor M/2 \rfloor x + y \bmod M$ is strictly optimal if $M \in \{1, 2, 3, 5\}$.

In conclusion, the contributions of this paper are twofold. For the efficient retrieval and displaying of geographically referenced data based on a two dimensional image or map, we derived strictly optimal assignments to multiple I/O devices and showed that these assignments satisfy the necessary and sufficient conditions for optimality. In the more general context of allocating multi-attributed data, our results give a complete solution for the important case of queries with arbitrary range on two attributes, and represent a significant step towards a general solution of the entire problem.

References

[AE93] K. A. S. Abdel-Ghaffar and A. El Abbadi. Optimal disk allocation for partial match inquiries. *ACM Transactions on Database Systems*, 18(1):132–156, March 1993.

[AU79] A. V. Aho and J. D. Ullman. Optimal partial-match retrieval when fields are independently specified. *ACM Trans. Database Syst.*, 4(2):168–179, 1979.

[CC92] C. Y. Chen and C. C. Chang. On GDM allocation method for partial range queries. *Information Systems*, 17(5):381–394, 1992.

[CLRS86] B. Chor, C. E. Leiserson, R. L. Rivest, and J. B. Shearer. An application of number theory to the organization of raster-graphics memory. *Journal of ACM*, 33(1):86–104, 1986.

[CR93] L. T. Chen and D. Rotem. Declustering objects for visualization. In *Proceedings of the 19th VLDB conference*, pages 85–96, August 1993.

[DS82] H. C. Du and J. S. Sobolewski. Disk allocation for cartesian product files on multiple-disk systems. *ACM Trans. Database Syst.*, 7(1):82–101, 1982.

[Du86] H. C. Du. Disk allocation method for binary cartesian product files. *BIT*, 26(2):138–147, 1986.

[Fal88] C. Faloutsos. Gray codes for partial match and range queries. *IEEE Transactions on Software Engineering*, 14(10):1381–1393, October 1988.

[FB93] C. Faloutsos and P. Bhagwat. Declustering using fractals. In *Proceedings of the Second International Conference on Parallel and Distributed Information Systems*, pages 18–25, January 1993.

[FBF+94] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3:231–262, 1994.

[FM91] C. Faloutsos and D. Metaxas. Disk allocation methods using error correcting codes. *IEEE Transactions on Computers*, 40(8):907–914, August 1991.

[GD90] S. Ghandeharizadeh and D. J. DeWitt. Hybrid-range partitioning strategy: A new declustering strategy for multiprocessor database machines. In *Proceedings of 16th International Conference on Very Large Data Bases*, pages 481–492, August 1990.

[GDQ92] S. Ghandeharizadeh, D. J. DeWitt, and W. Qureshi. A performance analysis of alternative multi-attribute declustering strategies. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 29–38, June 1992.

[KP88] M. H. Kim and S. Pramanik. Optimal file distribution for partial match retrieval. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 173–182, June 1988.

[LSR92] J. Li, J. Srivastava, and D. Rotem. CMD: A multidimensional declustering method for parallel database systems. In *Proceedings of the 18th VLDB Conference*, pages 3–14, August 1992.

[OS95] V. E. Ogle and M. Stonebraker. Chabot: Retrieval from a relational database of images. *Computer*, 28(9):40–48, September 1995.

[Smi96] T. R. Smith. A digital library for geographically referenced materials. *Computer*, 29(5):54–60, May 1996.

[Sun87] Y. Y. Sung. Performance analysis of disk modulo allocation method for cartesian product files. *IEEE Transactions on Software Engineering*, 13(9):1018–1026, September 1987.

[VH92] M. Vitterli and C. Herley. Wavelets and filter banks: Theory and design. *IEEE Transactions on Signal Processing*, 40(9):2207–2232, September 1992.