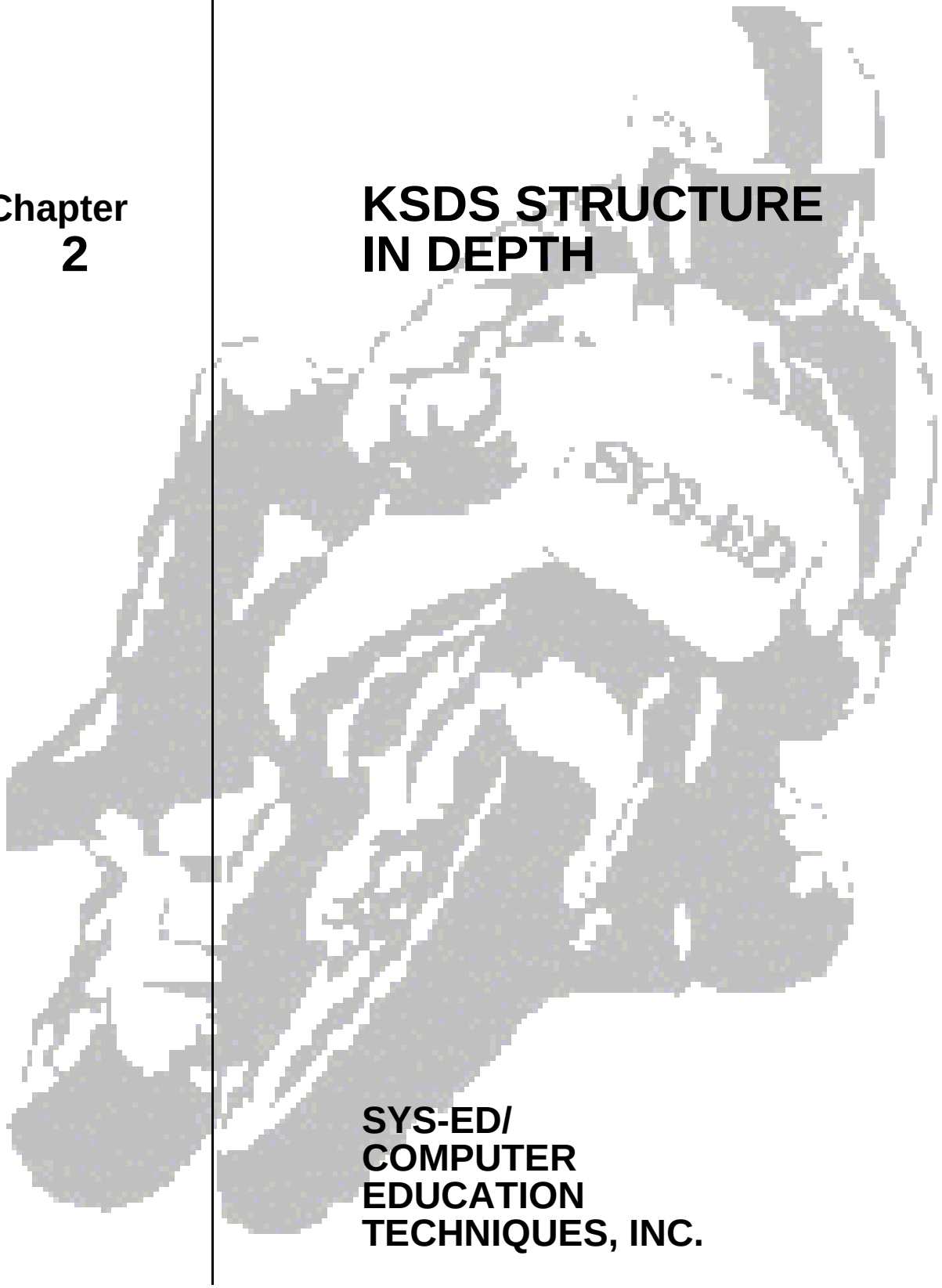
Chapter
2

KSDS STRUCTURE IN DEPTH

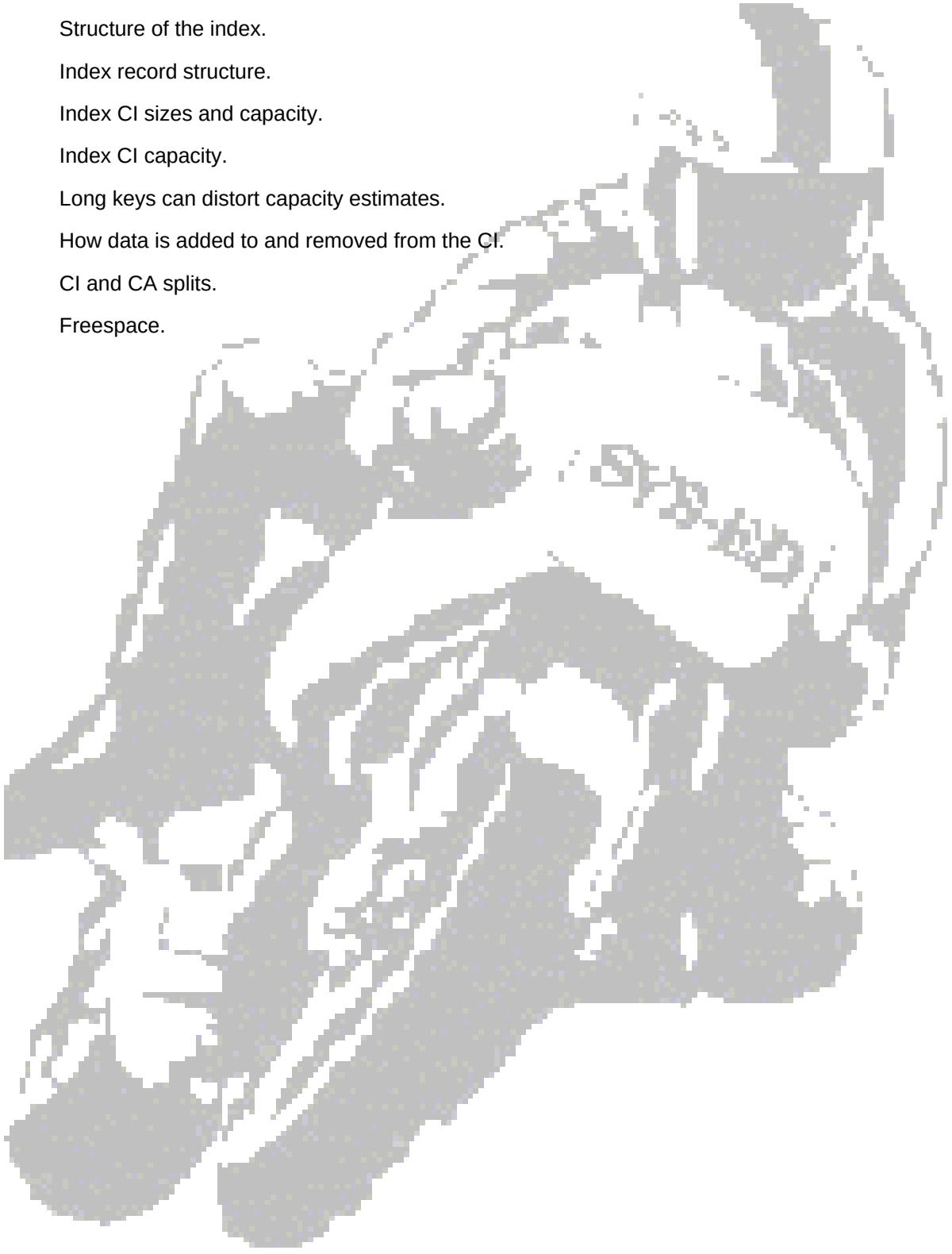


SYS-ED/
COMPUTER
EDUCATION
TECHNIQUES, INC.

Objectives:

You will learn:

- Structure of the index.
- Index record structure.
- Index CI sizes and capacity.
- Index CI capacity.
- Long keys can distort capacity estimates.
- How data is added to and removed from the CI.
- CI and CA splits.
- Freespace.



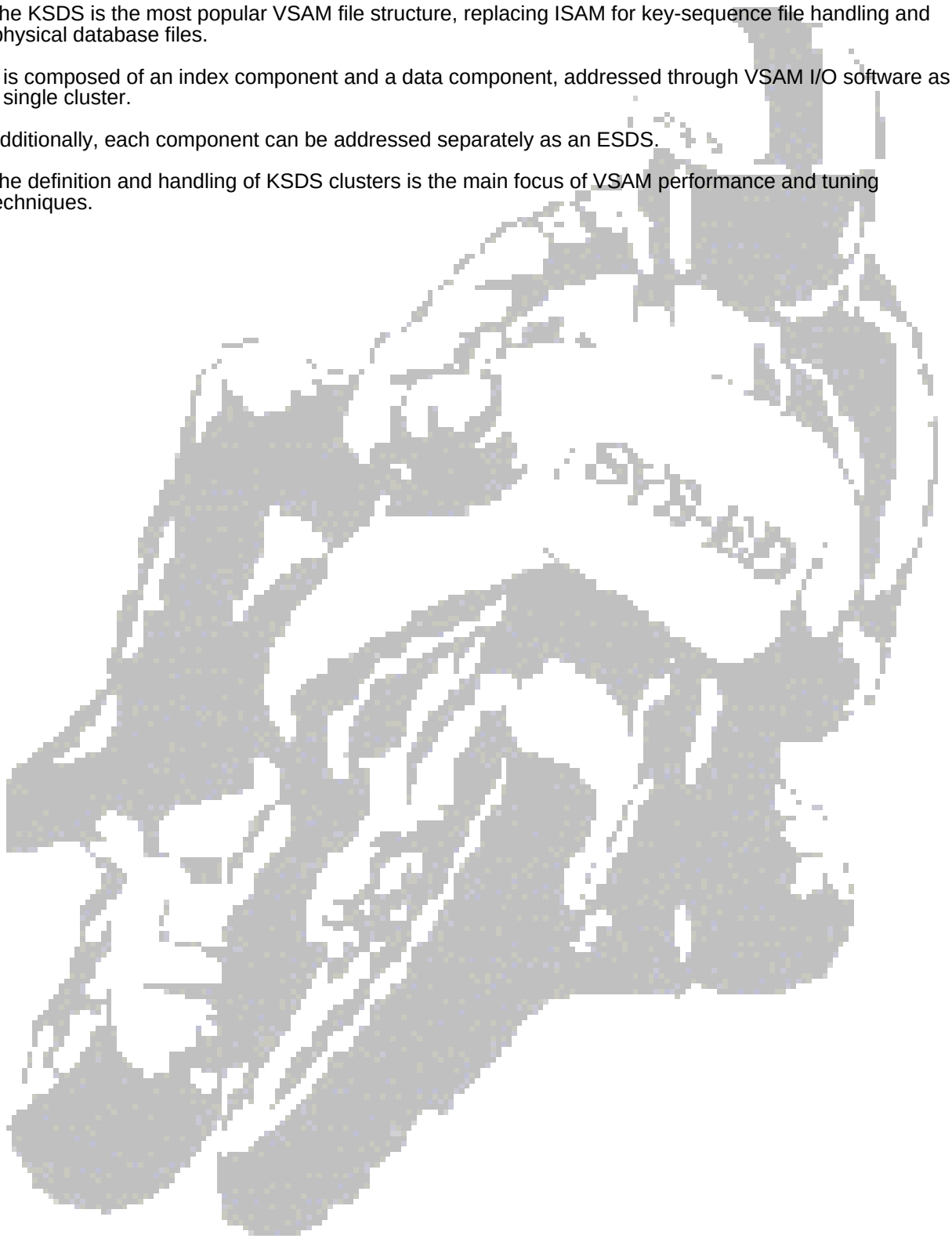
1 KSDS Structure in Depth

The KSDS is the most popular VSAM file structure, replacing ISAM for key-sequence file handling and physical database files.

It is composed of an index component and a data component, addressed through VSAM I/O software as a single cluster.

Additionally, each component can be addressed separately as an ESDS.

The definition and handling of KSDS clusters is the main focus of VSAM performance and tuning techniques.



2 A Structure of the Index

The index component of a KSDS contains information relating logical keys to data record locations in the data component of the cluster.

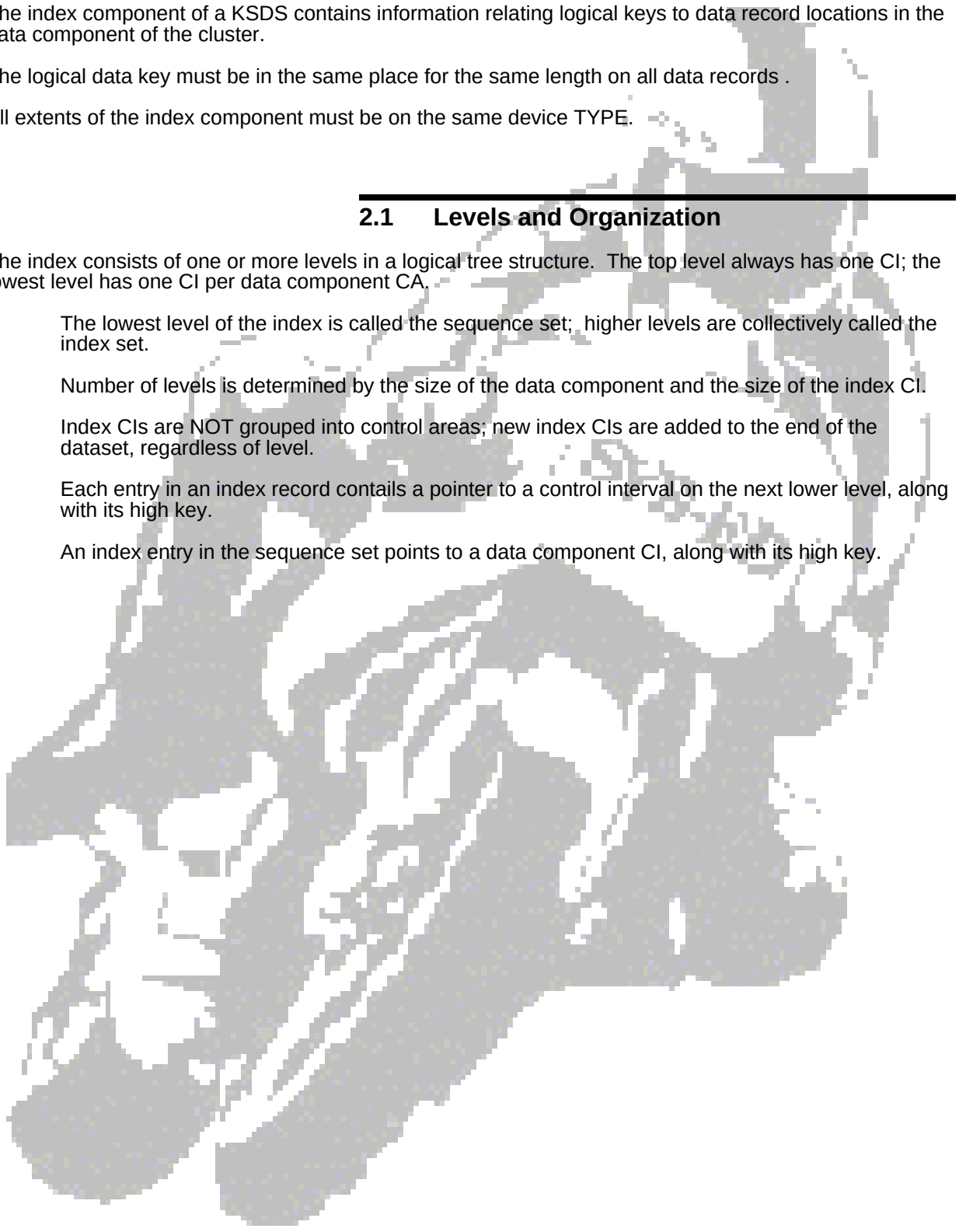
The logical data key must be in the same place for the same length on all data records .

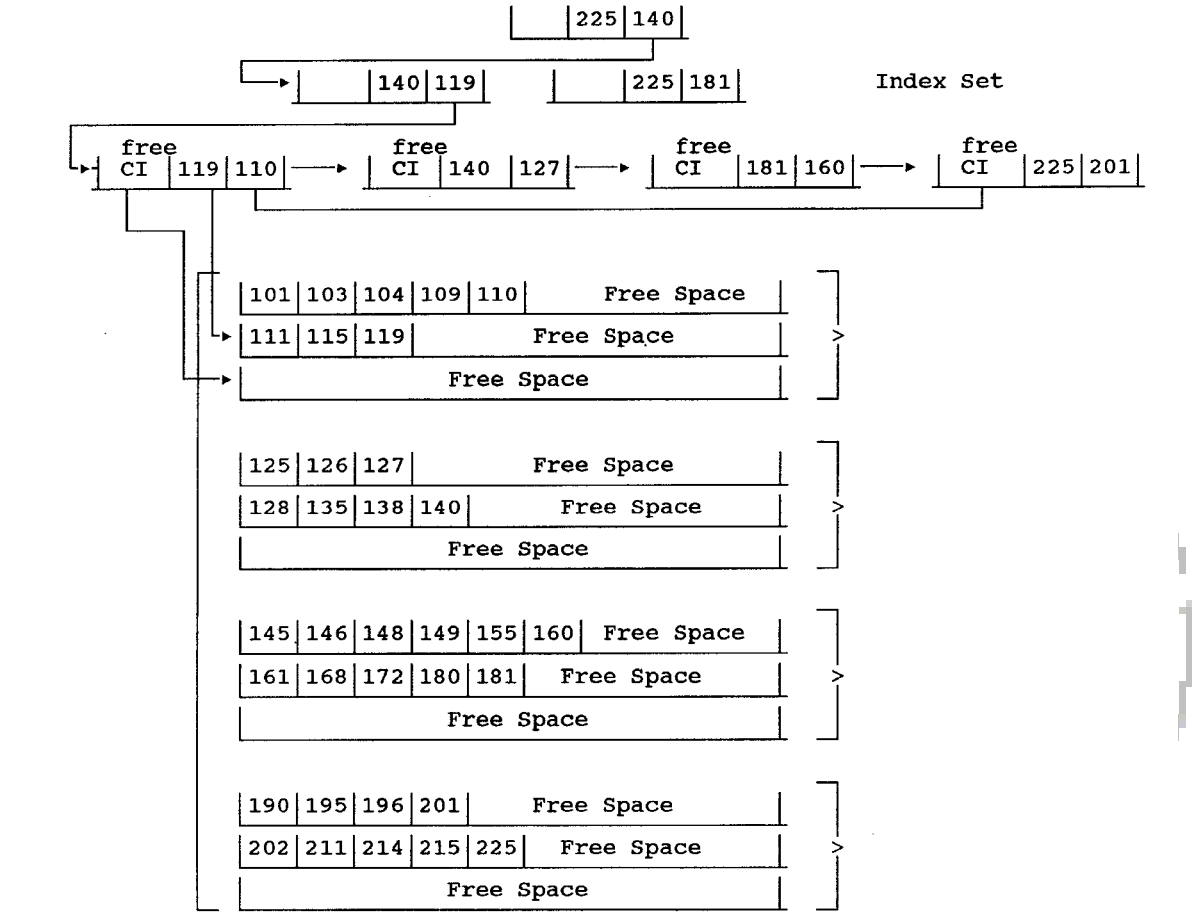
All extents of the index component must be on the same device TYPE.

2.1 Levels and Organization

The index consists of one or more levels in a logical tree structure. The top level always has one CI; the lowest level has one CI per data component CA.

- The lowest level of the index is called the sequence set; higher levels are collectively called the index set.
- Number of levels is determined by the size of the data component and the size of the index CI.
- Index CIs are NOT grouped into control areas; new index CIs are added to the end of the dataset, regardless of level.
- Each entry in an index record contails a pointer to a control interval on the next lower level, along with its high key.
- An index entry in the sequence set points to a data component CI, along with its high key.




225 140
140 119
225 181
Index Set
free CI 119 110
free CI 140 127
free CI 181 160
free CI 225 201
101 103 104 109 110 Free Space
111 115 119 Free Space
Free Space
125 126 127 Free Space
128 135 138 140 Free Space
Free Space
145 146 148 149 155 160 Free Space
161 168 172 180 181 Free Space
Free Space
190 195 196 201 Free Space
202 211 214 215 225 Free Space
Free Space

2.2 Index Record Structure

Each index record occupies a control interval by itself.

Index record	RDF : CIDF

Both the RDF and CIDF contain the record length, CI size - 7.

record length	0

The control interval after the last-used control interval in the index has a CIDF of all zeros.

Each index record consists of:

- Header.
- Pointers to free control intervals in the datacomponent (for sequence set records only).
- Unused space.
- Index entries.

	<---free ci pointers---->				<-index entries->		
header	pointer n	pointer 2	pointer 1	unused space	entry n	entry 2	entry 1

2.3 Index CI Sizes

Same CI sizes as permitted for data component.

Size can be specified in the VSAM Define or chosen by VSAM.

Index must be large enough to hold an entry to each CI in the data CA.

Smaller index ci size shortens search time through the index (if no extra I/O is created).

2.4 Index CI Capacity

Capacity of an index CI is related to:

- Index CI size.
- Length of the data key.
- Number of cis in the data ca.
- Effects of key compression routines.

For keys less than 150 bytes long, with average or better compression, the following rule of thumb is generally accurate:

# of Data CIs in CA	**Index Size**
# < 58	512
58 < # < 120	1024
120 < # < 184	1536
184 < # < 248	2048
248 < # < 312	2560
312 < # < 376	3072
376 < # < 440	3584
440 < # < 502	4096
.	.
.	.
.	.

3 Long Keys Can Distort Capacity Estimates

For a more accurate assessment, use the following formula:

$$\frac{\text{index size} - 31}{\frac{\text{keylength}}{2} + 5} = \text{\# of keys possible in size chosen}$$

Then:

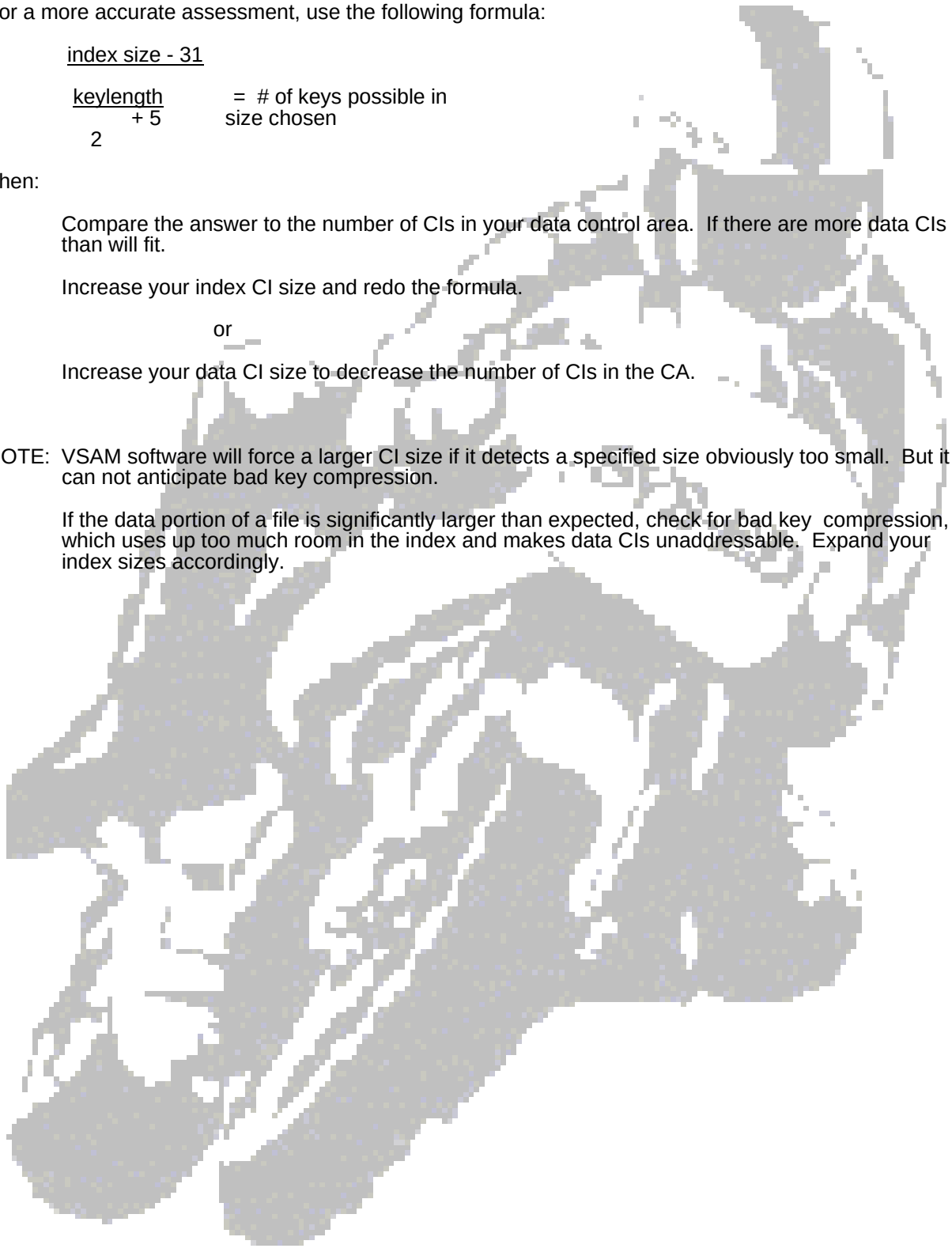
- Compare the answer to the number of CIs in your data control area. If there are more data CIs than will fit.
- Increase your index CI size and redo the formula.

or

- Increase your data CI size to decrease the number of CIs in the CA.

NOTE: VSAM software will force a larger CI size if it detects a specified size obviously too small. But it can not anticipate bad key compression.

If the data portion of a file is significantly larger than expected, check for bad key compression, which uses up too much room in the index and makes data CIs unaddressable. Expand your index sizes accordingly.



4 How Data is Added to and Removed from the Data Component

4.1 How Data is Added to and Removed from the CI

To keep a KSDS is sequence, new records are placed in the CI entitled to hold its logical key.

This is determined by finding the CI whose HIGH KEY as shown in the index is sequentially the first key higher than the new key.

If there is room in the CI for the new record (or extension of an old record), the CI is reorganized so that the new data is placed in the correct physical order.

- Records to the right of the insert location are moved farther right.
- RDF and CIDF values are updated to show the new record values and remaining free space.

Deletions also maintain the order within the CI.

- Records to the right of a deleted record are moved left to close the gap.
- RDF and CIDF values are updated to show the new record values and new free space.
- All insert and delete activity can cause RBAs of data records to change.

5 CI and CA Splits

When there is insufficient room in a CI to absorb additional data, a CI split will occur.

- An empty CI is located in the CA.
- A portion of the old CI is moved to the new one.
- The new data is added in it proper place.
- Indexes are adjusted to reflect the new data.

When no free CIs are in the CA, a CA split occurs.

- An empty CA is located.
- A portion of the old CA is moved to the new one.
- The CI split continues as usual.

CI and CA splits occur in two modes:

NIS: Normal Insert Strategy

- Used during direct mode processing.
- Splits occur at the half-way point of the CI or CA to be split, regardless of the insert location.

SIS: Sequential Insert Strategy

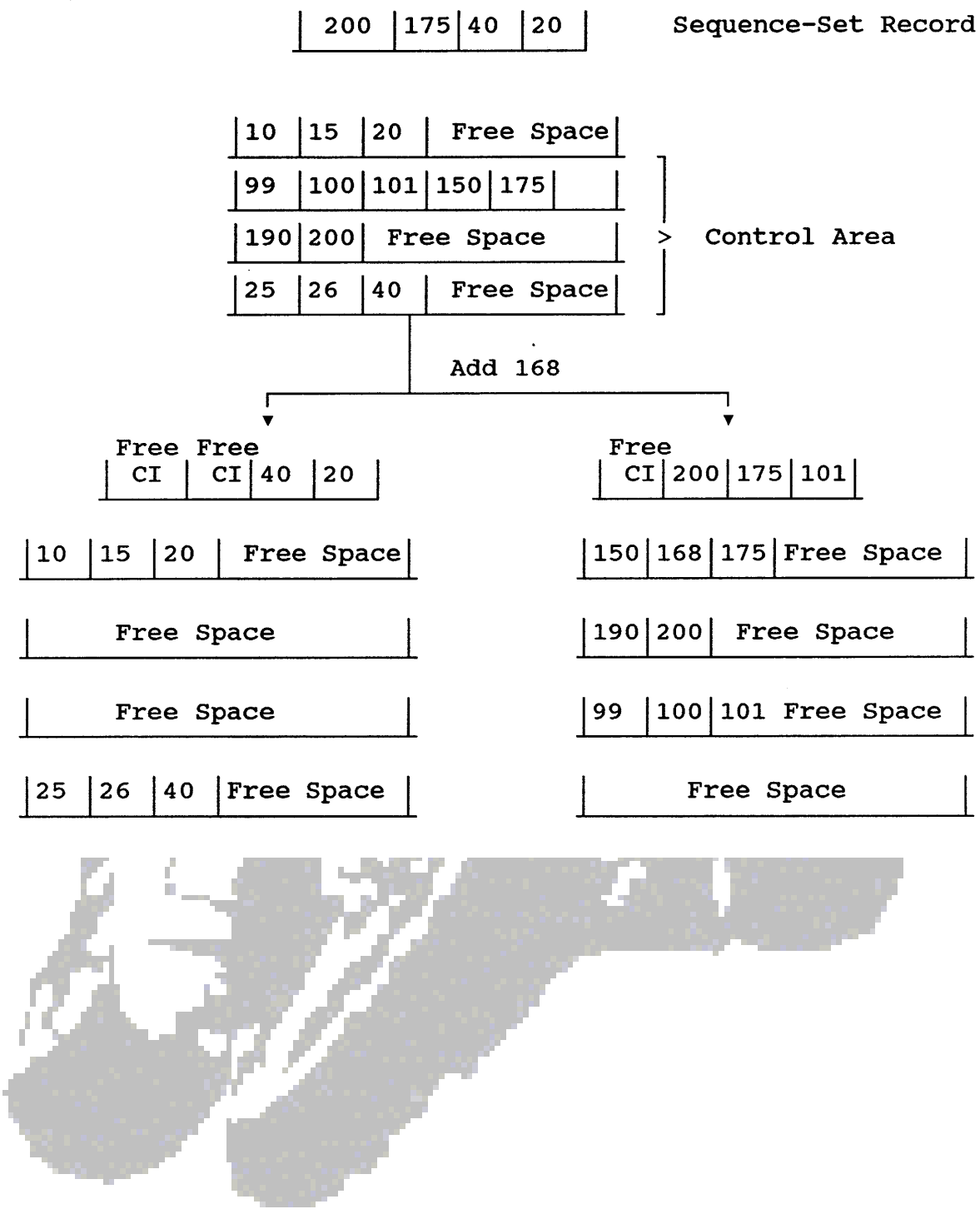
- Used during sequential mode processing, or when effective sequential processing is detected by VSAM software.
- Splits occur at the point of logical insertion of the new data.

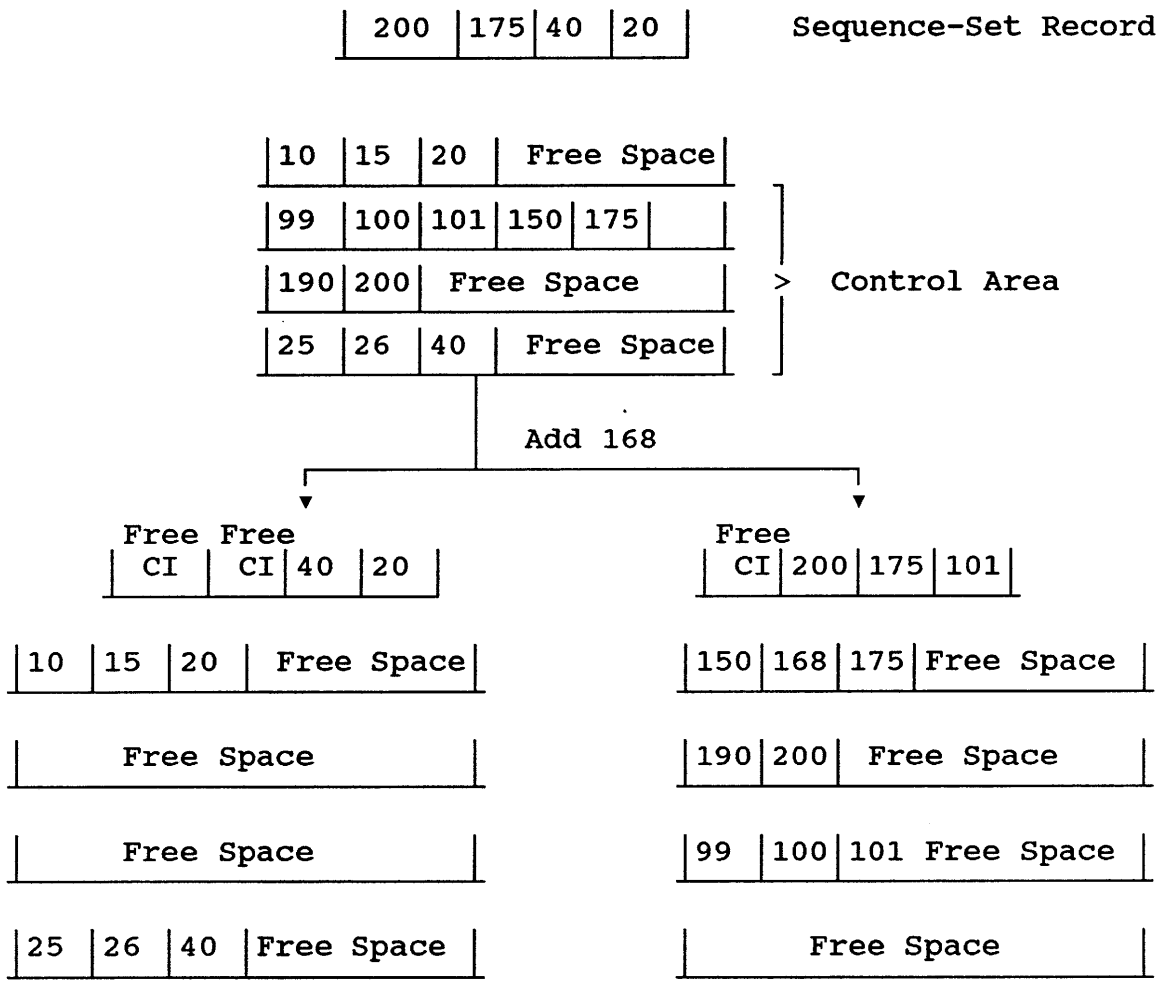
In the CI: If the key of the new record is higher than every key in the CI, it becomes the first record in the new CI. Otherwise, the new record is placed in the old CI, and all higher records go in the new CI.

In the CA: If the CI to be split is the highest key in the CA, it becomes the first CI in the new CA. Otherwise, it stays in the old CA, and all higher CIs are moved to the new CA. CI split logic takes over to choose the split location within the CI.

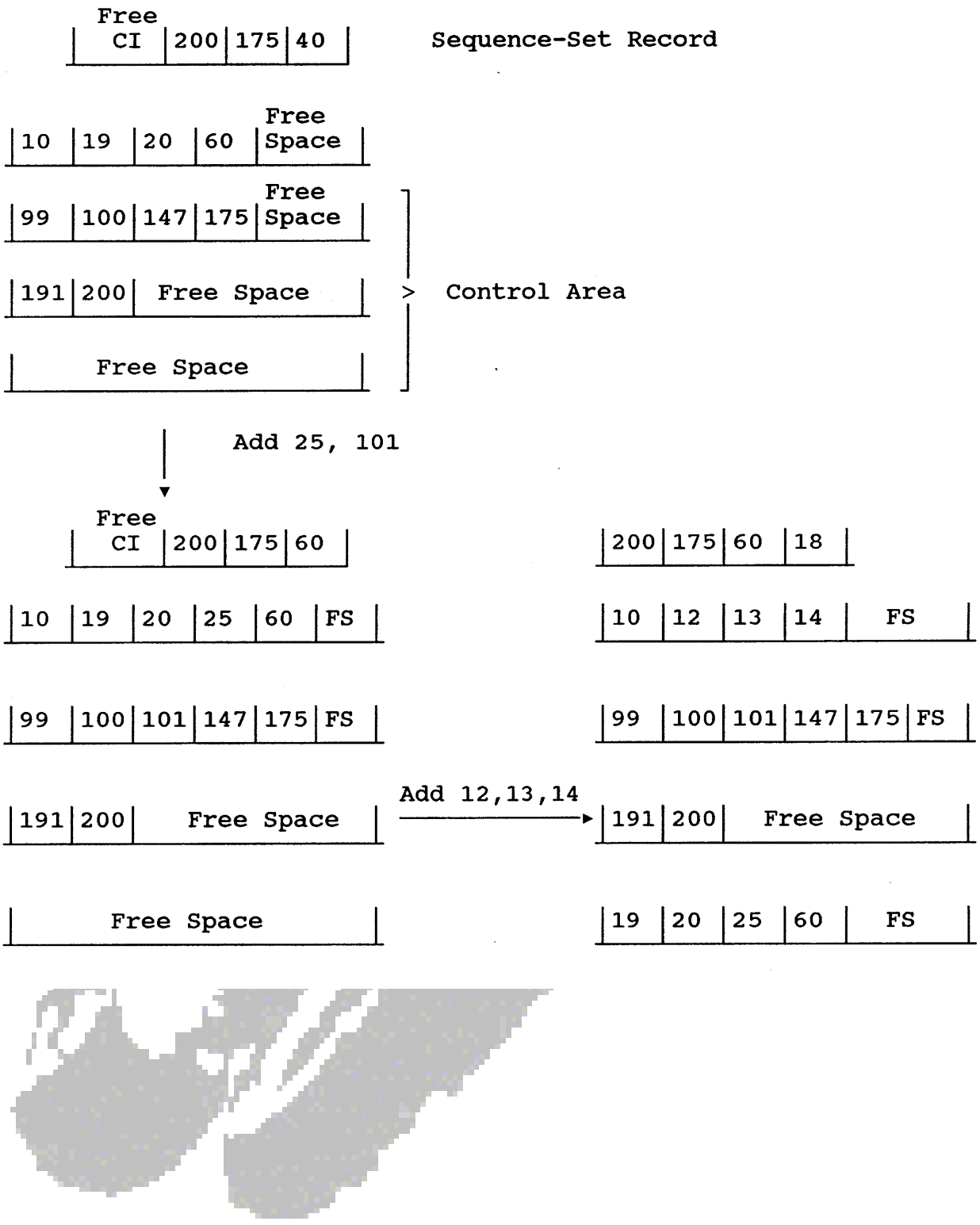
6 Key-Sequenced Datasets

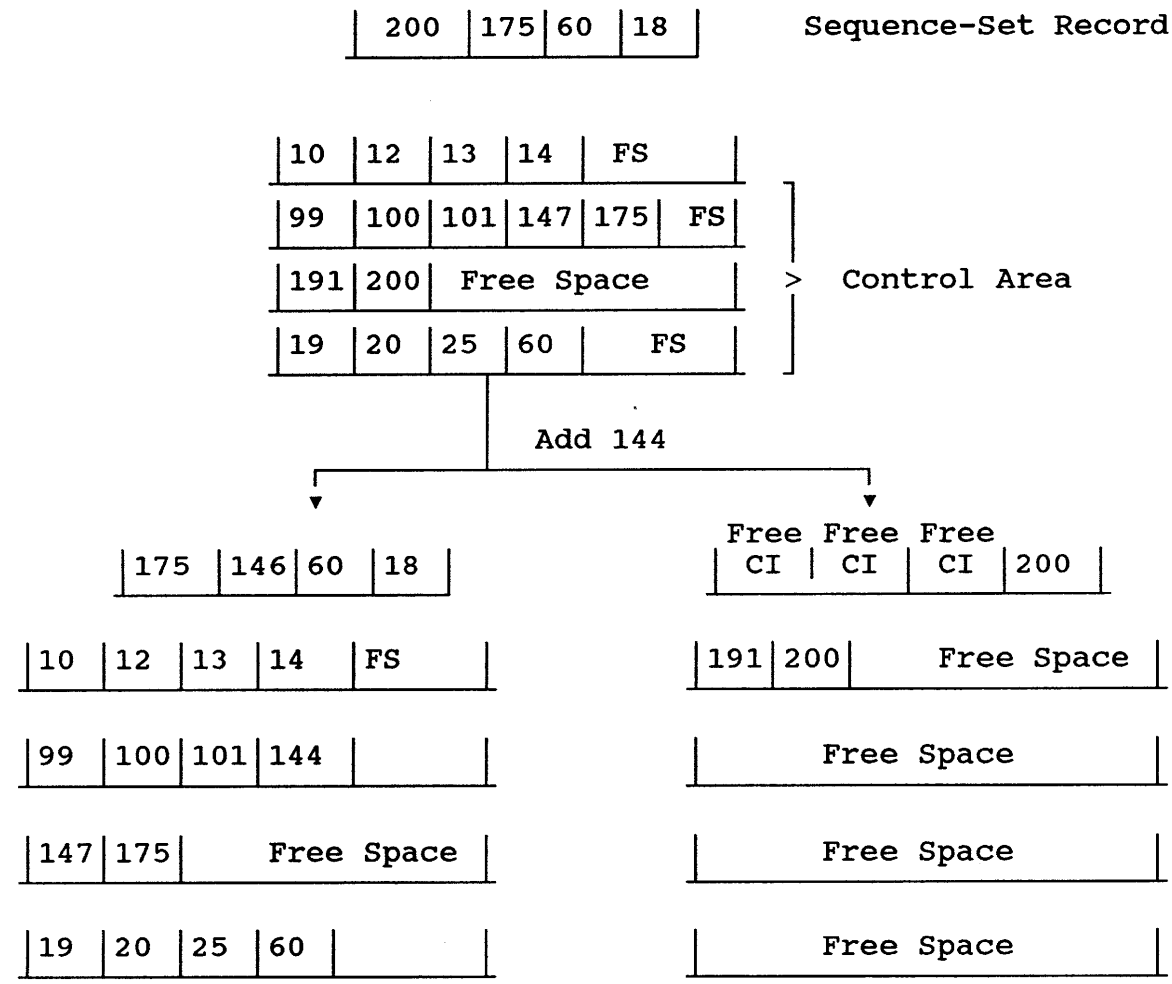
6.1 CI and CA Splits During Direct Processing




200 175 40 20
Sequence-Set Record
10 15 20 Free Space
99 100 101 150 175
190 200 Free Space
25 26 40 Free Space
Control Area
Add 168
Free Free CI CI 40 20
Free CI 200 175 101
10 15 20 Free Space
Free Space
Free Space
25 26 40 Free Space
150 168 175 Free Space
190 200 Free Space
99 100 101 Free Space
Free Space

6.2 CI and CA Splits During Sequential Processing




200 175 60 18
Sequence-Set Record
10 12 13 14 FS
99 100 101 147 175 FS
191 200 Free Space
19 20 25 60 FS
Control Area
Add 144
175 146 60 18
10 12 13 14 FS
99 100 101 144
147 175 Free Space
19 20 25 60
Free Free Free
CI CI CI 200
191 200 Free Space
Free Space
Free Space
Free Space

7 Costs of CI and CA Splits

7.1 I/O at Time of Split

Both CI and CA splits incur index I/O as well as extra Data I/O, since key ranges for CIs and CAs change as a result of the split activity.

In addition to normal retrieval and update -

For the CI split:

- A new CI is built and written out.
- The sequence set record is updated to add the new CI high-key, remove it from the free-ci list, and keep all CI high-keys in order in the index record.

For the CA split:

- A new CA is located by interrogating the catalog.
- Half the CIs in the original CA are read and written to the new CA.
- The sequence sets of both the old and the new CAs are rewritten to show the new contents of the CAs and reset the horizontal pointers in the Cas.
- The index record above the sequence set level is updated to show the new high-keys in the sequence set cis. If this results in a CI split at the index level above the sequence set, the I/O activity will be propagated upward until all changes are affected.

Note: If no space is available and the cluster is eligible for extension, a DADSM space assignment occurs. Both the catalog entry and the VTOC entry may be updated.

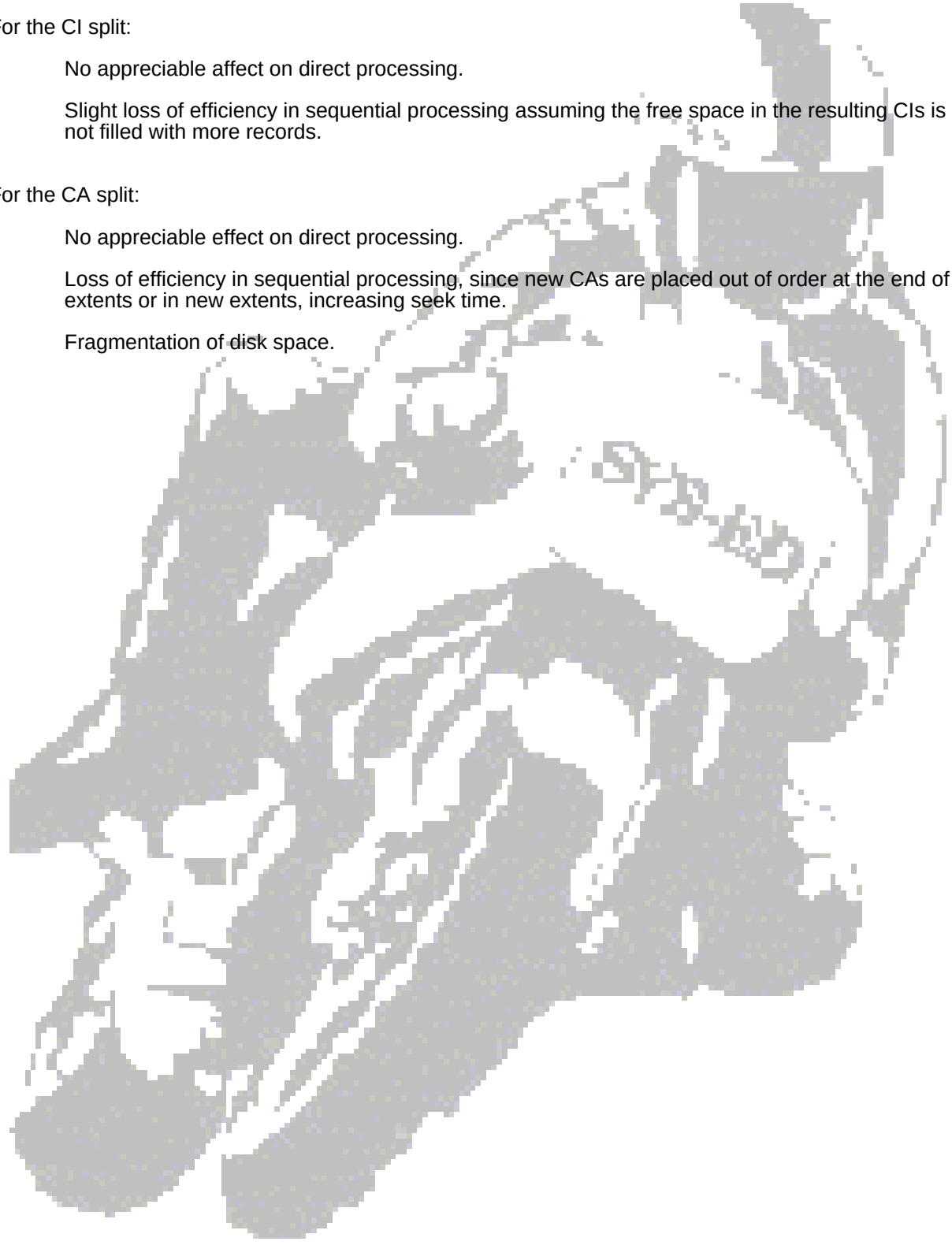
7.2 Costs After Split

For the CI split:

- No appreciable affect on direct processing.
- Slight loss of efficiency in sequential processing assuming the free space in the resulting CIs is not filled with more records.

For the CA split:

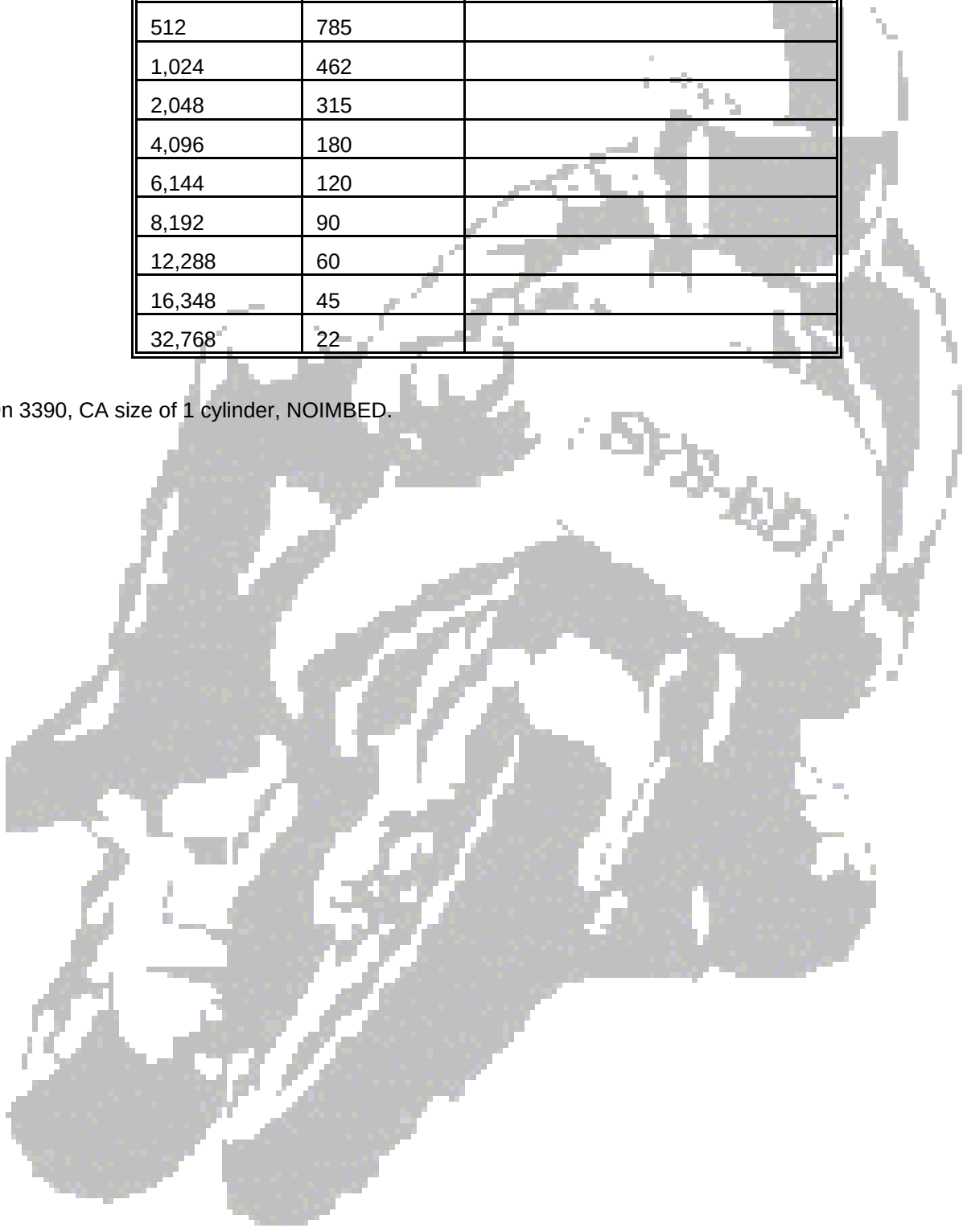
- No appreciable effect on direct processing.
- Loss of efficiency in sequential processing, since new CAs are placed out of order at the end of extents or in new extents, increasing seek time.
- Fragmentation of disk space.



7.3 CA Splits

CI SIZE	CI in CA	APPROX. I/O
512	785	
1,024	462	
2,048	315	
4,096	180	
6,144	120	
8,192	90	
12,288	60	
16,348	45	
32,768	22	

On 3390, CA size of 1 cylinder, NOIMBED.



8 The Concept of Freespace

User-specified amount of space, by percent, to be set aside at load time to provide for future direct access insert activity.

Specified at both the CI and CA level at define time.

Reestablished in a KSDS when file is reloaded with REPRO or IMPORT.

Used during NIS insert mode processing.

Set aside during SIS insert mode processing.