Title

Unit 037 - Fundamentals of Data Storage

Authors

037, CC in GIScience
Jacobson, Carol R.

Publication Date

2000

Peer reviewed

Unit 037 - Fundamentals of Data Storage

by Carol R. Jacobson, School of Earth Sciences, Macquarie University, NSW, Australia

---

Advanced Organizer

Topics covered in this unit

- This unit introduces the concepts and terms needed to understand storage of GIS data in a computer system, including:
    - the weaknesses of a discrete data model for representing the real world
    - an overview of data storage types and terminology
    - a description of data storage issues.

Intended Learning Outcomes

- After learning the material covered in this unit, students should be able to:
    - Recognise the differences between the data in a GISystem and the real world it represents.
    - Understand computer terminology as it applies to data storage.
    - Differentiate between different types of data storage.
    - Select different data storage types appropriate for various GISystem data.
    - Recognise the importance of data design in a GISystem.

Instructors' Notes

Full Table of Contents

Metadata and Revision History

Unit 037 - Fundamentals of Data Storage

- Computers are the enabling technology for GISytems, but their role is entirely the storage and manipulation of data.

- To fully understand the nature of the data stored in any GISytem, two issues are important:
    - the relationship between the stored data and the real world it depicts, and
    - the characteristics of data storage within computer systems

1. The Relationship between the Real World and Data in a GISystem

1.1 Weaknesses of a Discrete Data Model

- The real world is infinitely complex, BUT such infinite complexity cannot be depicted or processed within the bounds of a normal computing system.
- GISystems depict the world as being comprised of geometric objects:

  points, lines and areas for vector data models, and
  pixels for raster data models.
  This may or may not be an accurate depiction of reality. (See Unit 008 - Representing the Earth for more information.)
- In particular, the point, line and polygon model utilises objects with sharply defined boundaries. This sort of boundary is often not found in the real world.
    - For example, features such as wetlands are usually drawn on maps with sharp borders, although they have ill defined boundaries in the real world.
    - in the same way, many objects in GISystems have had sharp boundaries imposed upon them.
- In many ways the data in a GISystems give a simplified view of the real world. It depicts the real world, but has undergone three procedures:
    - selection,
    - representation in a standard way and
    - quantification.

1.2 Selection

- In a GISystem, selected aspects of the real world (those that are considered to be important), are included in the digital model of the real world.
    - Example: When the vegetation of a study area is included in a GISystem, the important areas of vegetation will be shown, and small areas such as copses or small clearings will be omitted.
    - Objects outside the selected study area are also considered unimportant, though in the real world their influence may be significant.
- The likely uses of the data will be decisive in determining which features are included.
    - The organisation collecting the data will have a responsibility in a particular area and the data will be chosen for this purpose.
    - Later users of the same data may fail to recognise inadequacies in the data, when it is used for purposes for which it was not designed.

1.3 Representation

- The real world objects that are included must be represented by an object defined in the GISystem software.
    - Example: A road network of a town comprises the road surfaces, footpaths, kerbing (and other structures). In a GISystem it will usually be represented by a network of lines defining the centre lines of the roads.
    - In a raster model GISystem, a connected series of cells (rather than a line) would represent a road.
- The objects that are represented in a GISystem will have defined boundaries.
    - Example: Real world features like forests or soil parcels, do not have sharp boundaries in the real world, however in a GISystem, they will be assigned boundaries.
- The likely uses of the data will again be decisive in determining the form of representation.
    - Example: At small scales, roads are usually represented by line networks defining the centre lines of the roads. For engineering uses, larger scales are employed, and the objects represented will include the kerbing and footpaths, the exact shapes of the curves, etc., but not just single lines. Thus, representation depends on usage, which also effects scale.

1.4 Quantification

- Computer systems generally store numeric values. Therefore numeric values are assigned to the characteristics of the real world which are included in the GISystem.
- This may be simple, or require considerable abstraction.
    - Example 1 (Simple): The maximum height of a mountain can readily be included in a GIS.
    - Example 2 (Complex): The height at an intermediate point on a hillside is can be represented in a number of ways:

      in the vector data model, heights can shown by the position of contour lines for set elevations.
      in the raster data model, an average height for each pixel can be recorded, or the height at the mid-point of each pixel.
    - Example 3 (Complex) There are many options for coding the characteristics of a forest in a GIS, including:

      numeric codes for forest categories such as rainforest or woodland;
      the canopy closure expressed as a percentage; or
      a numeric code for the dominant species of tree in the forest.
- A computer system stores unique or discrete values. These may or may not faithfully represent the continuum of values that exist in the real world.
- The nature of the data is important, as different types of mathematical operations can be performed on different data. Numerical values can be defined with respect to ***nominal, ordinal, interval*** or ***ratio*** scales of measurement.
    1. **Nominal**

On a nominal scale numbers merely establish identity.
No mathematical operations can sensibly be carried out on this data.
Example: Rain gauges within a study area may be given a numerical identity code. The identity numbers do not indicate any order in terms of rainfall at the site.

2. **Ordinal**

   On an ordinal scale numbers establish order only.
   Comparisons of size can be made, but no other mathematical operation.
   Example: Air pollution monitoring equipment situated in different suburbs will enable the suburbs to be rated 1st, 2nd, 3rd, etc according to their air quality. This information will not tell us how much worse the 5th and 8th suburbs are compared to the 1st.

3. **Interval**

   On interval scales the difference between numbers is meaningful, but the numbering scale does not start at zero.
   Subtraction makes sense but division does not.
   Example: If temperatures are measured at various locations, then it is sensible to say that 20°C is 10 degrees warmer than 10°C, not that it is twice as warm as 10°C.

4. **Ratio**

   On a ratio scale measurement has an absolute zero, and the difference between numbers is significant.
   Mathematical operation such as addition, subtraction, and division make sense.
   Example: The population data coded for census districts can be manipulated in many ways, in particular the population can be dived by area (another ratio scale measurement) to obtain population density.

- A knowledge of the real world that the GISystem is attempting to model is essential to make informed decisions on the optimum data storage formats.

- More information on the representation of the real world in a spatial database is available in Unit 030 - Abstraction and incompleteness

2. Storage of Digital Data within a Computer System

Understanding the elements of computer storage will enable a GIS user to design optimum storage for different types of data.

2.1 Bits

- Computers function on two basic elements, on and off.
- The smallest processing unit is called a ***bit*** (short for Binary digIT). Each bit can have one of 2 values: "on" (indicated by the value 1) and "off" (indicated by the value 0).

- Bits are grouped together in sets of eight, called *bytes.*

2.2 Binary Systems

- Computers use a ***binary system*** for storing numbers. In a binary system, the only figures are 1 and 0.
- Binary systems are best explained by comparison to the familiar decimal system. (A decimal system is uses 10 figures.)
    - In a decimal system, the digits 206 represent the number that is made up of

      2 lots of 10^2 plus 0 lots of 10^1 plus 6 lots of 10°
      (from high school mathematics: 10^2 is 100, 10^1 is 10, and 10° is 1)
    - In a binary system the digits 101 represent the number that is made up of:

      1 lots of 2^2 plus 0 lots of 2^1 plus 1 lot of 2°
      (2^2 is 4, 2^1 is 2, and 2° is 1, so the number is 4 + 1 = 5)
- Counting from 1 to 10 in binary gives the following series of numbers:

  1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010
- Binary and decimal systems are just 2 number systems: potentially there are many others that could be used. Two others, octal and hexadecimal, are common because they are also used in computing. Table1 gives the numbers for counting from 1 to 20 in these systems.

2.3 Bytes

- One ***byte*** of storage is 8 bits, and so can hold integer numbers in the range 0 to 255.
    - (Integer numbers are numbers that don't have decimal points.)
    - The number 255 is the limit, because it is the binary number 11111111 (8 1's)

      which equals 1 (which is 2°), plus 2 (which is 2^1), plus 4 (2^2),
      plus 8 (2^3), plus 16, plus 32, plus 64, plus 128.
- This is a very useful range of data. Much (but certainly not all) non- spatial data in a GIS, falls in this range. For example:
    1. even in complex forested areas, tree species can usually be allotted a discrete code within this range.

       BUT elevation often falls outside this range.
    2. Remote sensing data is designed to fall into this range for ease of transmission from the sensors in the satellite back to earth.
- Integer data with values greater than 255, require more than one byte of storage to be stored in a computer system. See Storage of Numerical Data

2.4 The ASCII Coding System

- The ***ASCII*** (pronounced ass-key) coding system is another important use of bytes of data. The acronym stands for American Standard Code for Information Interchange.

- Every letter and number key on a keyboard has a unique code.
    - Seven bits are used to give 127 code numbers which are assigned to each key (upper and lower case letters have different codes).
    - There are also codes for special characters such as Tabs and Carriage Returns. See Table 2 - ASCII codes for Various Keyboard Characters
    - The 127 basic ASCII codes can be extended to by using the eigth bit to give a total of 255 codes. These extra codes include characters for international (non-English) characters such as , mathematical symbols such as , and graphics characters such as those in table borders.
    - Recently a 16-bit international character set, that allows kanji and chinese characters, has been introduced.
- Textual data in a GISystem are stored as ASCII characters. See Storage of Character Data
- There are ASCII codes for numbers as well as letters. These codes are completely different from the binary representation of the numbers.
    - The 4 in the name "42nd Street" has the ASCII code 52 which is expressed in binary as 00110100,
    - the number 4 in a binary system is 00000100.
- ASCII coding is an important standard for the transfer of data.
    - The way numerical data is stored in a computer depends on the architechture of the computer (see Storage of Numerical Data). Data created on one type of computer will be mis-interpreted by a computer with a different architechture. Data should be converted to ASCII before transfer, as all computers correctly interpret ASCII codes.
    - most GISystem software offer "export" options which produce ASCII files.
    - the disadvantage to ASCII coding, is that GIS data files are very much larger coded this way.

2.5 Storage of Numerical Data

- The way numerical data is stored in a computer depends on the architechture of the computer: this depends on the type of computer ("personal" or mainframe) and the age. The number of bits that the computer uses as the basic unit to store data is called the ***word size***. For example, the following sizes are commonly used:
    - 16-bit (2-bytes) "personal computers" (previous generation)
    - 32-bit (4-bytes) "personal computers" (current generation)
    - 64-bit (8-bytes) mainframes
- Computers store negative numbers by the use of a ***sign bit***.
    - The sign bit is usually the high order bit, that is the bit in the left-most position.
    - If this bit is set (has the value 1) the number is negative, if it is unset (has the value 0) the number is positive.
    - The software indicates to the computer processor whether the high order bit is to be treated as a sign bit, or part of the number. (Except for byte data, numeric data is usually stored as signed data.)
- In a GISystem most spatial data, which may be in decimal degrees or UTM coordinates, will include data with decimal places. In computer systems this is usually called ***floating point*** data.

Each number is stored in two parts:
  - the first part (called the mantissa) is the value of the number,
  - the second part (called the exponent) is the power of ten (or the power of two in a binary system) to multiply the mantissa by, to obtain the original number.
- This sort of storage is more complex, but it can be illustrated with a fairly simple example:

  The latitude of Sydney in decimal degrees is 33.86167 S.
  Using decimal numbers, the numbers used to store this figure would be 3386167 and 2.
  - the number 3386167 is the mantissa,
  - it is assumed to have a decimal point on its immediate left, so its value becomes .3386167
  - the number 2 is the exponent, because .3386167 must be multiplied by 10^2 (or 100) to obtain the original number (33.86167).
  - (The same steps are used in a binary system, but the numbers are different.)
- In general, integer data storage has advantages over floating point storage:
  - In choosing a storage type, users should consider, the intended uses of the data stored in the GISystem, and the types of values that will need to be represented.
  - Depending on the GIScience software being used, floating point numbers may require considerably more storage space than integer numbers, because two numbers must be stored for every value.
  - Similarly floating point numbers are more complex to process.

2.6 Storage of Character Data

- Character data stored in a GIS may be single letters or characters (for example * or a space), single words, or groups of words such as a property owner's name or vegetation species.
- Groups of letters or characters are usually called ***character strings***.
- Numbers can be stored as character data. For example it is useful to be able to store lot numbers for land parcels.
  - A number stored as a character string will be stored as a series of characters.
  - It is not usually possible to use numbers stored as character strings in mathematical operations, such as addition.
- If a character string includes spaces (the ASCII code 32), it is necessary to use a ***terminator*** to indicate the extent of the string. Different software uses different terminators, for example some use single quotes (') and others use double quotes (").

---

3. Design of GIS Data for Efficient Storage

3.1 Storage Terminology

- The following terminology is commonly used to describe storage capacity of various data storage devices:
  - **K** stands for kilobyte or approximately 1000 bytes (actually 1024 bytes)

  - **M** stands for megabyte or approximately 1,000,000 bytes (actually 1,048,576 bytes)
  - **G** stands for gigabyte or approximately 1,000,000,000 bytes (actually 1,073,741,824 bytes).
- Design of GISystem data should ensure that storage capacity is used efficiently.

3.2 Efficient Use of Data Storage Capacity

- Spatial accuracy is important in GIS, so it is inevitable that large storage demands are made for storing spatial data. Most GISystem software allow the user little (or no) choice on how spatial data, such as UTM co-ordinates or decimal degrees, are stored.
- For non-spatial data, such as cell values in the raster model or polygon or line attribute code in the vector model, design is important, and most GISystem software is flexible in storage types for this data.
  - the floating point number 4.0 will be allotted a minimum of 4 bytes of storage space,
  - if the integer number 4 is an equal representation of the real world value of the data, then one byte of storage is all that is needed.

  Therefore storage requirements can be more than doubled by inappropriate choice of data type.
- The raster model for GISystem data, typically requires more storage than the vector model, due to the large number of cell values that must be stored, so design issues are more important with this model.
- Design of data storage is important for two reasons:
  - Users of GISystems inevitably find that requirements for data storage expand at least as rapidly as the capacity of available storage devices, so efficient use of available apace is essential.
  - When data is processed it must be read from the storage device and after processing be re-written. Reading and writing data, is usually the slowest part of data processing. If poor design has resulted in the use of unecessarily large amounts of storage, processing time will be slowed, by the reading and writing of redundant storage bytes.
- The characteristics of different data storage types are summarised in Table 3.

- More information on design of databases can be found in Unit 045 - Non-spatial database models and Unit 050 - Fundamentals of information science.

4. Summary

- Data in a GISystem depicts the real world, but the complexities of reality have been altered in at least three ways:
  1. selection of a subset of real world objects,
  2. convertion of real world objects to standard data types with specifc boundaries, and
  3. quantification or the use of discrete numerical values, to represent complex

characteristics.

- The smallest processing units in a computer system are bits. Bits are grouped together in sets of eight to form bytes. One byte can store a positive integer between 0 and 255, or the code for any keyboard character.
- ASCII codes are numeric codes for each letter, number and special character on a keyboard: each code can be stored in one byte. Textual data in a GISystem is stored as ASCII codes. To ensure data integrity, data should be converted to ASCII codes before it is exchanged between different GISystem sites.
- Numeric data in a GISystem can be stored as integer or floating point data types. In GISystems coordinate data usually has decimal places, and so is stored in a floating point data type.
- Inefficient use of storage space, not only reduces storage capacity, but also slows processing times.

5. Review and study questions

(Questions 1 & 2 are from the Original Core Curriculum)

1. Compare the data storage needs of:
    1. the data transmitted per year by the EOS satellites, which generate 1 Terabyte (10^{12} bytes) per day;
    2. the US Bureau of the Census's TIGER files of street networks, which are about 10G (gigabytes) and are updated every 10 years; and
    3. a database of 100M (megabytes)created for use in a one-time environmental impact study.
2. "User expectations about data volumes rise at least as rapidly as the capacity of available storage devices" Discuss.
3. Recommend data storage types for storing raster data for the following GISystems data:
    1. remote sensing data;
    2. nominal data for planning zones in a local government area, a total of 20 codes are used to indicate different zones;
    3. elevation data for a national park, which includes rugged terrain up to 2000 metres;
    4. geological codes, for example: Qal (alluvium), Tea (volcanic agglomerate), Pi (shale), Psn (quartz sandstone), and Cig (granite).
    5. data on the pH of soil samples, the range of values is 6.5 to 8.5 (data is recorded with one decimal place);
    6. data from a habitat study which record the locations of various tagged animals;
    7. canopy heights in a forested area, data is recorded in metres with 2 decimal places.
4. A variety of coding schemes are used to convert non-ASCII data to ASCII equivalents before electronic transmission. Find the names of some of the common methods.

6. References

6.1. Print references

- Bernhardsen, Tor (1992) *Geographic Information Systems*. Arendal, Norway: Viak
- Burrough, Peter A. & McDonnell, Rachel A. (1998) *Principles of Geographic Information Systems*. Oxford University Press, Oxford.

Cartography texts provide useful insights into representing the world abstractly.

- Campbell, John (1984) *Introductory Cartography*. Prentice Hall, New Jersey.
- Muehrcke, Phillip C. and Muehrcke, Juliana O. (1992) *Map Use: Reading, Analysis and Interpretation*. JP Publications, Madison, USA.

There are numerous computer books written for lay readers, which provide information on digital data storage. Two readable books for non-computer specialists are:

- Trainor, Timothy N. & Krasnewich, Diane (1996) *Computers!*. McGraw-Hill, New York.
- Radlow, James (1995) *Computers and the Information Soceity*. Boyd & Fraser, Danvers, USA.

Citation

To reference this material use the appropriate variation of the following format:

> Carol R.Jacobson. (1998) Fundamentals of Data Storage *NCGIA Core Curriculum in GIScience*, http://www.ncgia.ucsb.edu/giscc/units/u037/u037.html, posted October 6, 1998.

Created: February 24, 1998. Last revised: June 29, 1998.

Fundamentals of Data Storage in a GISystem

Instructors' Notes

Syllabus Context

- This unit describes the storage of real world data in a digital format.
- It would be best included in a teaching program after units which provided insights into modelling the real world, see GISCC section Representing the earth.
- Issues concerned with how the real world is sampled are also related, see GISCC section Abstraction and incompleteness

Background Information

One of the key words suggested for this module was "discreteness". This is one of a number of important characteristics of digital data. When real world information is stored digitally, a number of changes in its nature are determined by the technology. In addition to characteristics enforced by the processes of abstraction, generalisation (etc.), which will be covered in other units, this unit highlights the processes of selection, representation and quantification, because these processes are required by the technology. Students should be aware of all these processes and their potential effects on real world data.

An underlying theme of this unit is that understanding the elements of computer storage will enable a GIS user to nominate sensible storage and processing procedures for different types of data. The design of data storage, requires knowledge that all numbers are not identical in a computer system. Therefore Table 3 is essential to the theme of the unit, although following the NCGIA guidelines it is not placed in the body of the unit text.

The description of binary number systems is only important as background information. Most students with a geography background need not understand binary numbers: most students with a computing background will already understand them. Tables 1 and 2 are peripheral to the theme of this unit, and merely provide additional information on the existence of non-decimal number systems and ASCII codes.

Demonstrations and Exercises

Most GISystem software enables users to specify data types for new data sets (Erdas IMAGINE is particularly flexible). It is interesting to create data sets with the same data but different data formats, and compare file sizes: this could be set as a student exercise.

Changes in storage requirements are larger with raster than vector files (so IMAGINE provides striking examples). Changes with vector data are also observable: remind students that 2 kbytes on a 100 record test data set will be 2 Mbytes on a 100000 record real life data set.

The suggested study questions emphasise the application of knowledge of data storage formats.

Unit 037 - Fundamentals of Data Storage

Table of Contents

Unit 037 - Fundamentals of Data Storage

Metadata and Revision History

1. About the main contributors

- Carol R. Jacobson, School of Earth Sciences,

  Macquarie University, NSW, Australia

2. Details about the file

- unit title
  - Fundamentals of Data Storage
- unit key number
  - 037

3. Key words

4. Index words

5. Prerequisite units

6. Subsequent units

7. Other contributors to this unit

8. Revision history

- Created: February 27, 1998
- Revised: March 02, 1998
- Revised following reviewers comments: June 29, 1998

Back to the Unit.

Table 1. Counting Numbers in Other Number Systems

Notes

1. If mankind had evolved with 3 fingers (and a thumb), we would have probably used an octal number system for counting, that is one based on 8 figures.
   The 8 figures used in Table 1 are: 0, 1, 2, 3, 4, 5, 6, 7.

2. The characteristics of computer processors made the hexadecimal number system widely used: this system has 16 figures.
   Table 1 uses the commonly used figures, which are: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Decimal System	Binary System	Octal System	Hexadecimal System
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14

Table 2. ASCII codes for Various Keyboard Characters

Character	ASCII Code	Character	ASCII Code
(space)	32	0	48
!	33	1	49
"	34	2	50
#	35	3	51
$	36	4	52
%	37	5	53
&	38	6	54
'	39	7	55
(	40	8	56
)	41	9	57

Character	ASCII Code	Character	ASCII Code
A	65	a	97
B	66	b	98
C	67	c	99
D	68	d	100
E	69	e	101
F	70	f	102
G	71	g	103
H	72	h	104
I	73	i	105
J	74	j	106

Table 2. A Summary of Data Storage Types

Type	Byte¹	Integer	Non-integer	Character
Other Names	char 8-bit		numeric decimal float	string
Examples	6 200	402 -10	7.5 3.14159	Hawkesbury '99 Balaclava Rd'
Invalid Examples (reason why)	310 (value is too large) 2.1 (must not have decimal point)	3.2 (must not have decimal point) 27m (only figures allowed)	6/2/1998 (only figures allowed)	Bondi Beach (terminators, which are usually quotes, are required around names with spaces)

Notes

1. Byte data is frequently used for raster data (especially remote sensing imagery), but seldom used for vector data.
2. In some software 4 byte (32-bit) storage is used for integers.
3. With non-integer storage, 8 byte storage is used for storing coordinate data in some vector software. This may be referred to as ***double***.