IxD Theory 2: Telecomunicazioni

IUAV University of Venice

Visual and Multimedia Communication graduate programme

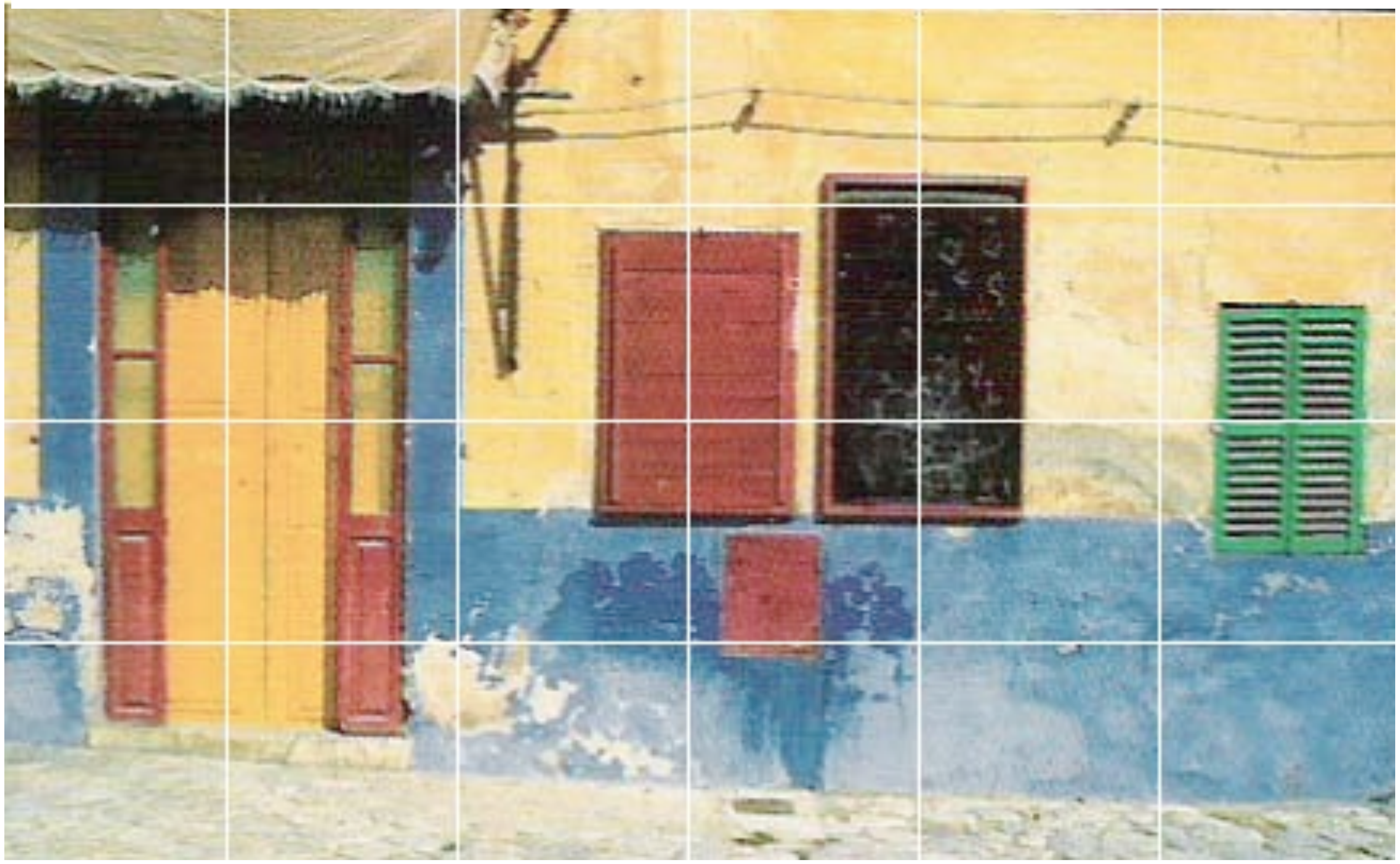
Digitization and representation

Part 2

Summary: representation 1

- The computer can only manipulate numbers
- Things in the world which we want to manipulate by computer must be translated into numbers
- Images are digitized by the computer into an array of pixels
- Information about the colour of pixels is stored in video memory
- The computer's video controller scans the video memory and signals to the screen to display the correct colour for each pixel

EXERCISE: PIXELLATE AN IMAGE



Problems of sampling (quantisation)

The computer only deals in discrete (digital) measurements, not continuous (analog) ones we find in real life.

In the exercise we sampled the picture at very coarse intervals. This meant that

1) often squares were almost equal between one colour and another so information about the colour not chosen was lost.

2) the final image was far from the original.

Finer sampling helps but information is always lost.

Digitization and representation

Bit-mapped images

Vector images

Sound

Text

IMAGES

Have you noticed that you can’t do some things in Photoshop that you can do in Illustrator and vice versa?

Have you heard the phrase

*‘**bit-mapped graphics**’ or*

*‘**vector graphics’**?*

Representation: images

Photoshop and Illustrator store information about the image in different ways.

Photoshop is primarily a **paint** program, and Illustrator a **vector drawing** program.

Bit-mapped images (paint programs)

Paint programs store the image as an array of pixels.

E.g: If you draw a black line, when you have made the final click, the program switches to black all the pixels that make up the line.

The program remembers only the colour of the pixels in the array. That is why you can't move the line later.

Vector graphics (e.g. Illustrator, Flash)

Vector drawing programs store the image as a list of operations. It then draws these elements into the video memory.

	start		*end/radius*		*line/fill*	*line*	*colour*
	x	*y*	*x*	*y*	*type*	*thickness*	
line	100	100	200	200	dotted	2pt	red
circle	100	100	20	-	solid	1pt	black
line	200	200	250	200	dotted	2pt	red
rect line	100	100	250	250	solid	1pt	black
rect fill	100	100	250	250	70%	-	green

(Properties, like ‘dotted’ or ‘red’, are also stored as numerical codes.)

EXERCISE: VECTORIZE AN IMAGE

In 1922 the artist Lázlo Moholy Nagy designed some paintings using gridded paper. He then – by phone and only using words and numbers – instructed a factory to make the paintings.

This exercise asks you to do something similar.

Bit-map graphics versus Vector graphics

Bit-map graphics

- Uses a lot of memory to store image, even with **compression**
- Scaling images needs **interpolation**
- Can easily do image processing operations (e.g. changing colour balance or tone)
- Can retouch parts of the image

Vector graphics

- Stores image economically (good for web)
- Can scale image without losing quality (important for print)
- Image elements remain editable

3D images: objects in a scene

Two types of representation:

Surface model – represents just the outside edges and surfaces.

Good for architecture.

Solid model – represents internal as well as external characteristics.

Good for industrial objects.

3D images: objects in a scene

Uses x,y,z coordinates to represent each vertex of the object.

X Horizontal

Y Vertical

Z Depth

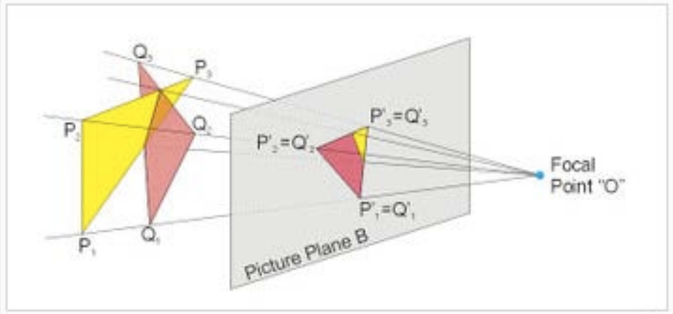
Stored as a list as in drawing programs.

3D images: objects in a scene

The program calculates the perspective of the scene, the light and shadow falling on the objects and the qualities of the surfaces, in relation to the eye of the viewer.

It can simulate a 'fly-through' as if a viewer was walking or flying through a scene.

For very complicated scenes every frame of the fly-through can take hours or days to compute.



Perspective Projection

Image from Wikipedia- http://en.wikipedia.org/wiki/3D_computer_graphics

TEXT (e.g. MS Word)

Text

A text is represented by numbers which correspond to letters of the alphabet, figures and punctuation. It is stored as a continuous stream of characters and formatting commands.

It is displayed on screen or by printer using bit-mapped or outline fonts.

A text is translated for printing using Postscript, a language that translates the word-processor information to highly refined instructions for printing.

Text

ASCII text coding: 7 bits, 128 numbers, letters, punctuation. Only the roman alphabet.

Unicode text coding: originally 16 bits, 65 536 codes. Now nearly 100 000 different characters are encoded. Used for writing systems worldwide.

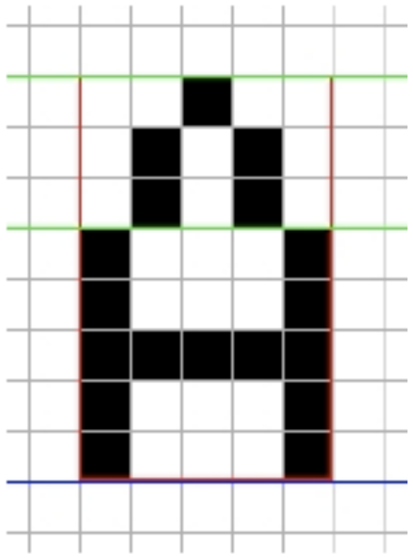
http://www.pcguide.com/res/tablesASCII-c.html

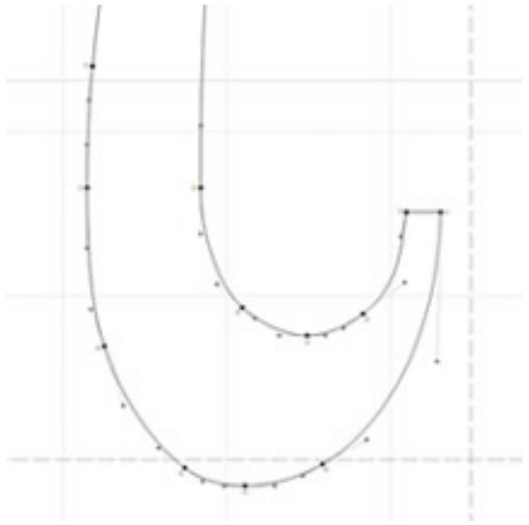
http://www.unicode.org

Text: bit mapped/outline fonts

Bit-mapped fonts are designed for a particular sized pixel matrix. They have to be designed separately for each size.

Outline fonts store their letters as scalable outlines which are then filled.



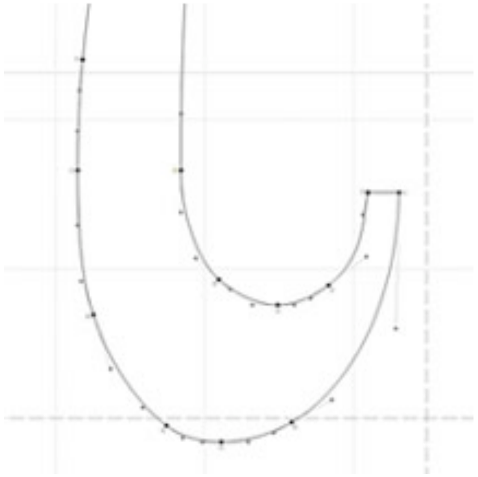


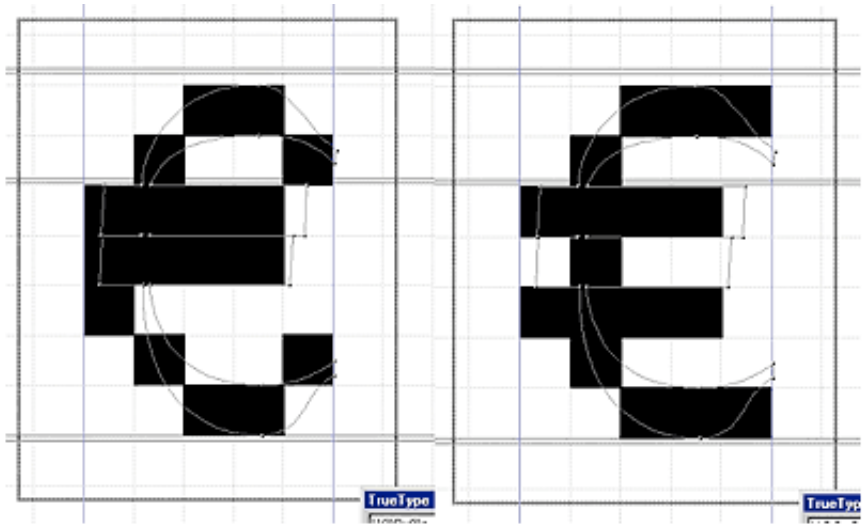
http://www.linotype.com/2712/ancientfonts.html?PHPSESSID=052763dbd3688ef5741d1c614767866b

Text: bit mapped/outline fonts

Like with drawing programs, little is lost when the characters are enlarged or reduced for print.

BUT when they are rendered on the screen there are not enough pixels to render them accurately, especially at small sizes.

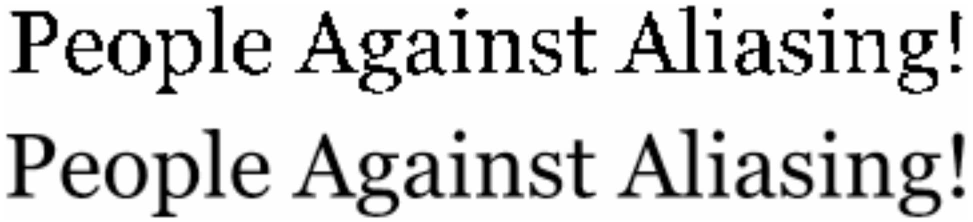


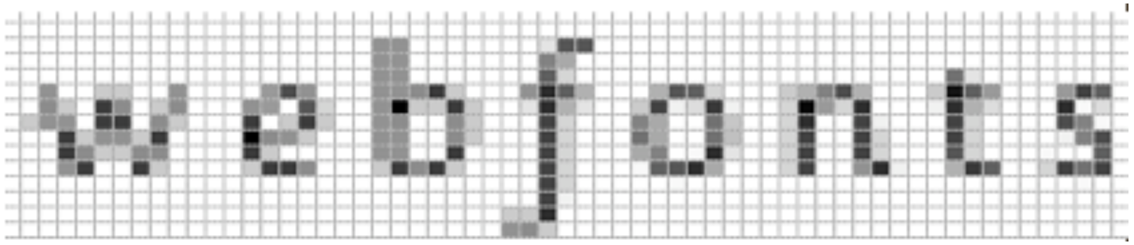


http://www.linotype.com/2712/ancientfonts.html?PHPSESSID=052763dbd3688ef5741d1c614767866b

Text: bit mapped/outline fonts

Fonts can be **anti-aliased** to make them appear smoother: the pixels surrounding the letters are averaged in colour between the background and the character.





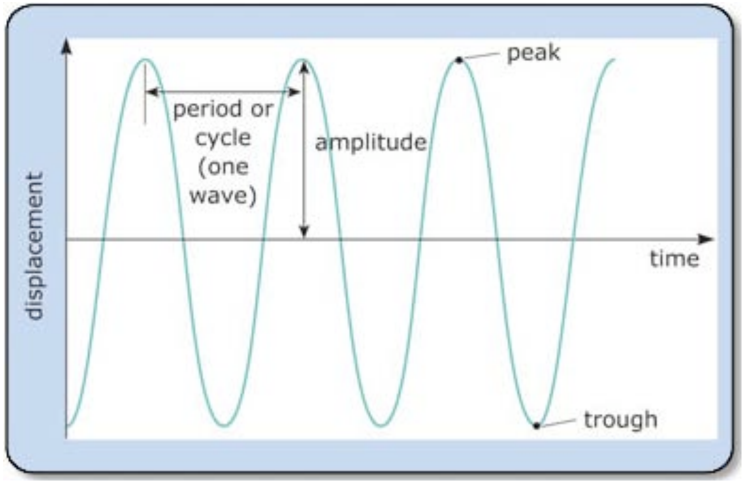
http://www.will-harris.com/webtype/anti-alias.html

SOUND

Sound

Sound is represented by a graph of the wave displaced by the vibration of the sound.

It has **frequency** (the number of troughs and peaks in a given interval, usually a second) which indicates the **pitch** of the note.



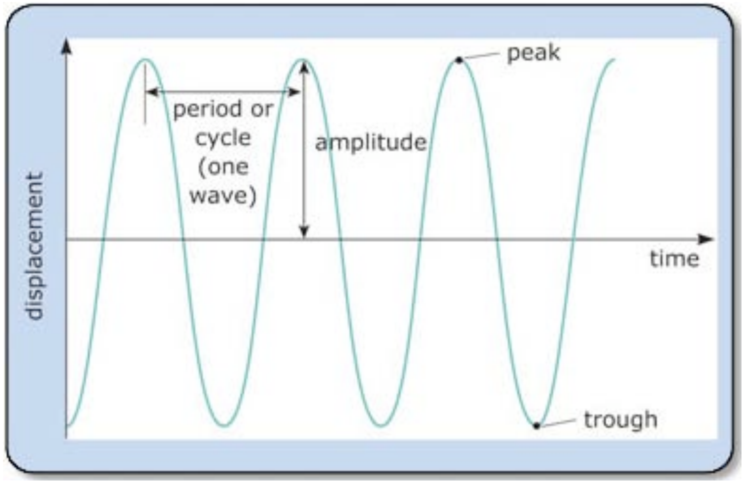
http://openlearn.open.ac.uk/mod/resource/view.php?id=32600

Sound

Sound is represented by a graph of the wave displaced by the vibration of the sound.

It has **frequency** (the number of troughs and peaks in a given interval, usually a second) which indicates the **pitch** of the note.

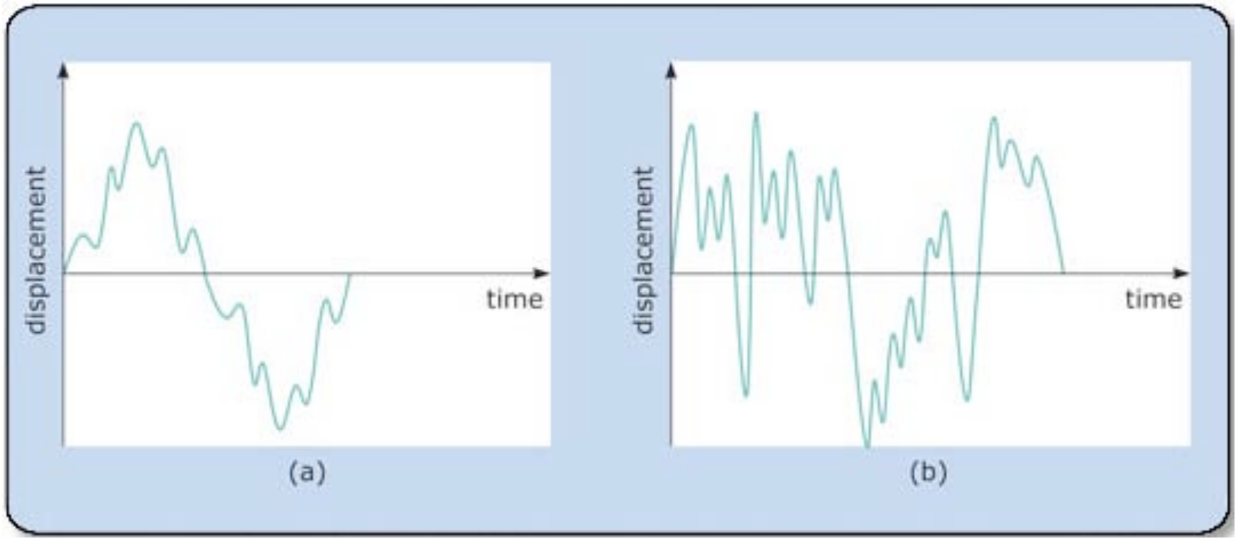
and **amplitude** (the height of the peaks) which indicates **loudness**.



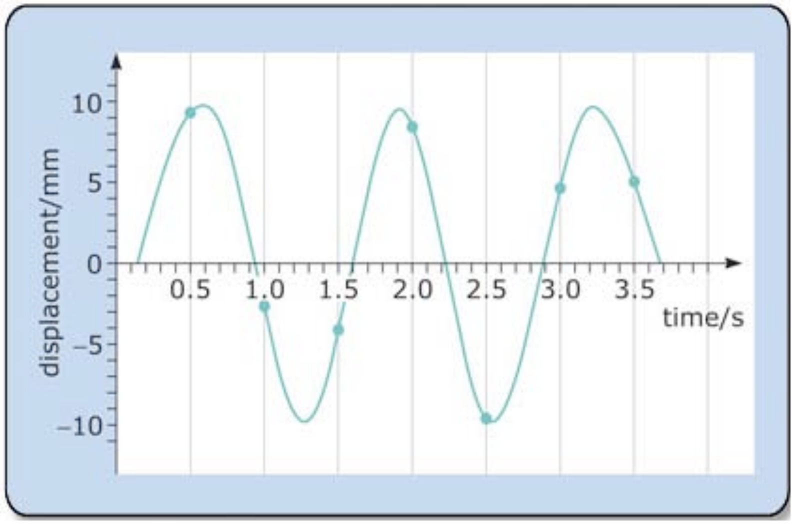
http://openlearn.open.ac.uk/mod/resource/view.php?id=32600

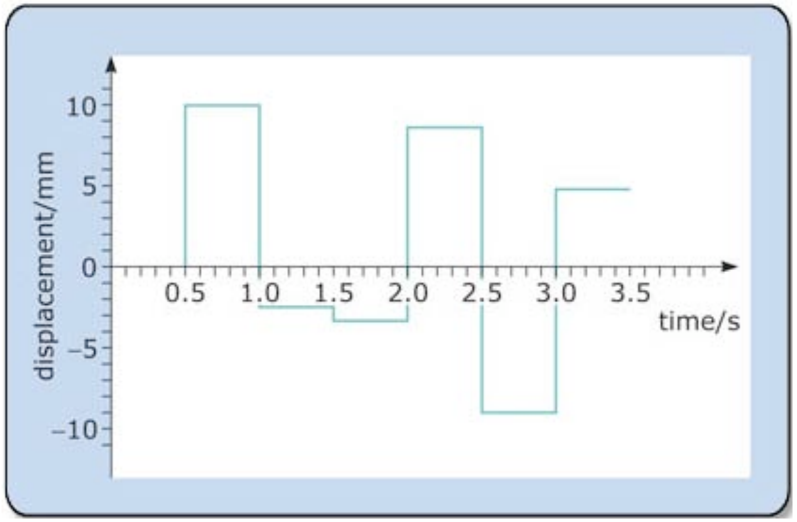
Sound

Waveforms can be very different:



Sound

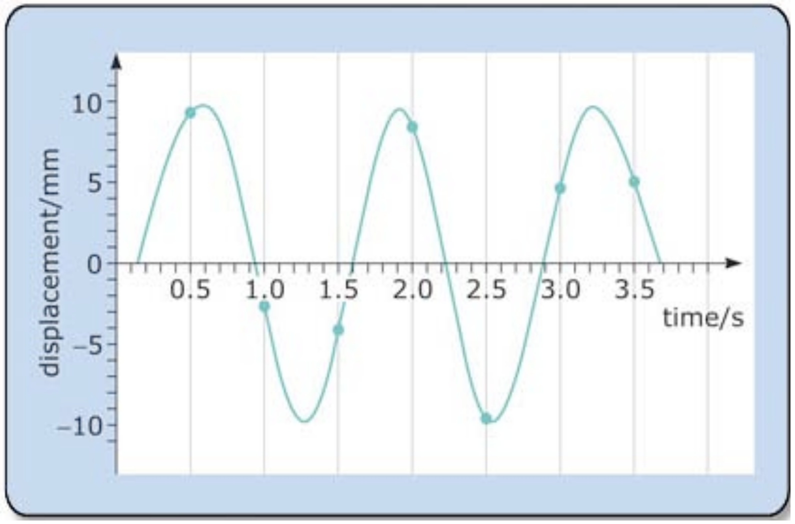


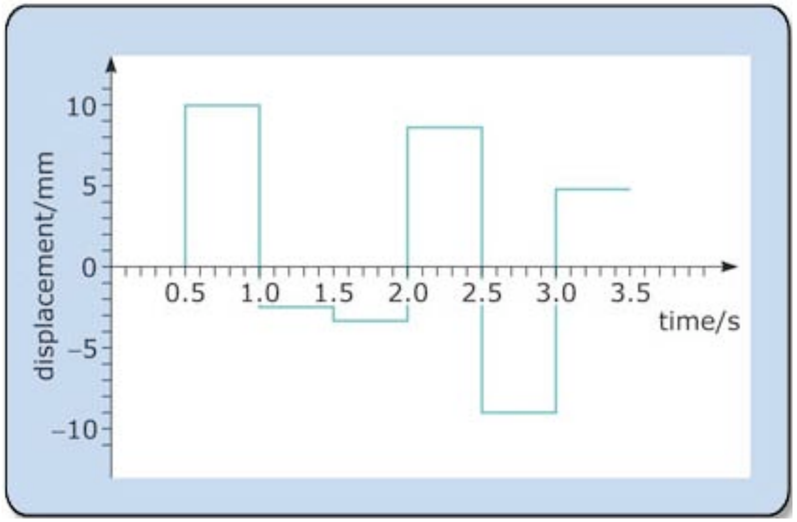


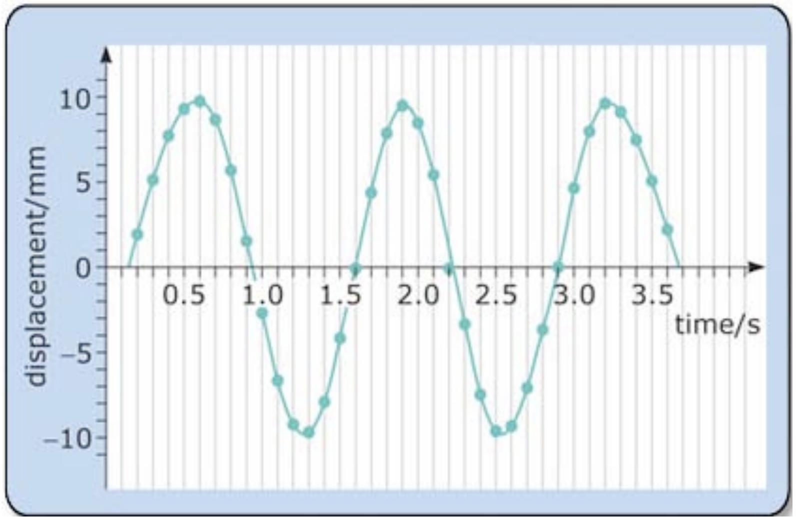
To translate sound into numbers for the computer, we sample (measure) the sound at intervals.

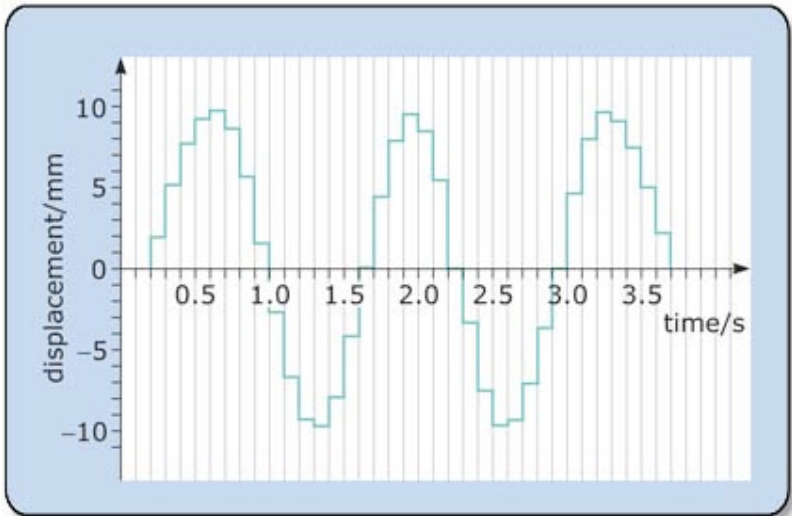
Here the sound is measured every .5 of a second. We plot this again, changing the analog wave into discrete, digital steps.

Sound









Sampling every 0,1 of a second: closer to the analog

Sound

The computer receives the sound by microphone and an analog-to-digital converter converts the sound into digital steps.

The computer may then process the sound, or just store it, and play it back in reverse: a digital- to-analog converter outputting through speakers.

Sound: statistics

Bit depth: the number of bits to represent the each sample of the wave.

16 bits gives 65 536 levels.

24 bit gives 16 777 216 levels

Frequency: the number of times per second you sample the wave. Good fidelity needs sampling at twice the highest frequency of the sound.

A CD is sampled at 44.1 hertz (44 100 samples per second).

(http://www.tweakheadz.com/16_vs_24_bit_audio.htm)

Sound

Storage: audio is stored as a continuous stream of bytes. At a bit depth of 16 bits (2 bytes) and sampling at 44.1 hertz (44 100 times per second), you will need 88 200 bytes per second. (88kb).

Say a song is 3 minutes long:

2 bits x 3 mins x 60 secs x 44 100 bytes =
47 736 000 bytes = around **50 megabytes** per song.

This can be reduced by compression techniques (e.g. MP3).