Discrete Methods of Picture Generation

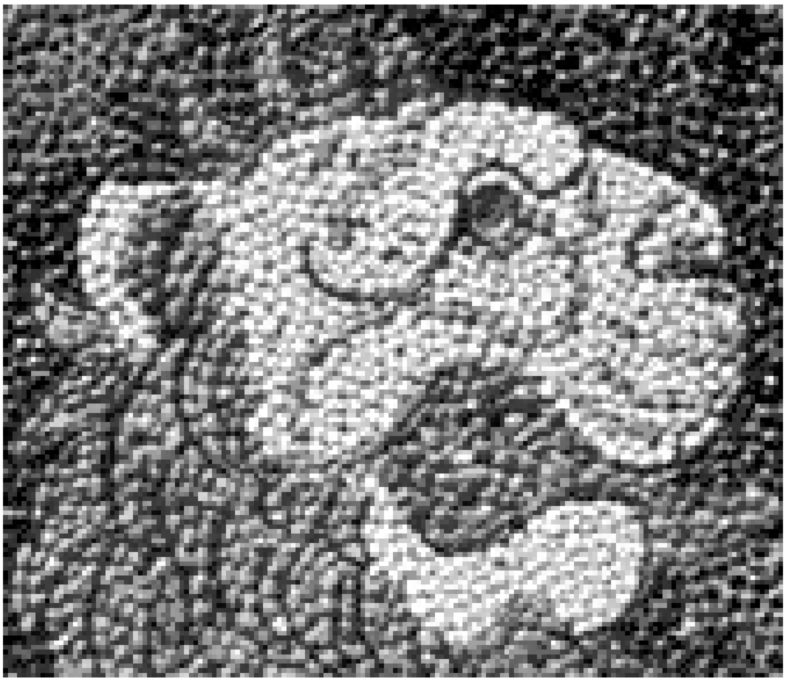
frequently used in both art and technology

(i) pointillistic paintings: dots can be as small as 1/16 of an inch in diameter (e.g., paintings of Australian Aboriginies, or of Georges Seurat)

(ii) gridded paintings (e.g., by Chuck Close or Salvador Dali)

(iii) (cross) stitching on canvas

(iv) pebble mosaics or tiled floors



A Greek pebble mosaic, detail from “The Lion Hunt” in Pella, Macedonia, circa 300 BC.

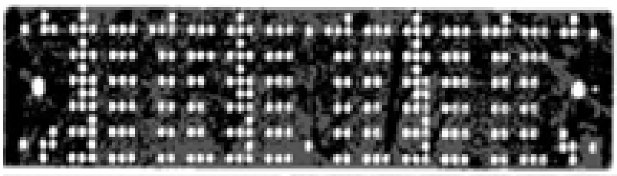
(v) patterns on fabrics or rugs

1725: B. Bouchon invents the idea of controlling a loom by perforated tape

1738: Falcon, a master silk weaver in Lyons, France, applies for an English patent on his automatic card-controlled loom

early 19th century: J.M. Jacquard greatly improves the design of card-controlled looms; thousands of Jacquard looms were soon in operation

Note: Pictures were generated by a programmed machine even before the first programmed machine (invented by C. Babbage) performed calculations on numbers!



Surviving example of a pattern woven by a Jacquard loom: a black-and-white silk portrait of Jacquard himself, woven under the control of a “program” consisting of about 24,000 cards (one is shown on the left).

Digital Pictures in 2005

Picture resolution: display parameter, dots per inch (dpi), standard for recent screen technologies is 72 dpi

printers: standard resolution is 300dpi to 600dpi

pictures typically captured with traditional (digital) cameras on planar surfaces: the light-sensitive array (CCD matrix, charge-coupled devices) is a rectangular set of square cells in a plane (each cell of a few square micro millimeter)

panoramic cameras (pictures are captured onto a cylindrical surface) are an example of an alternative camera design

picture size: beginning of 2005 already up to 10 Megapixel per captured picture for cameras priced below NZ$1,000

Small and medium sized cameras seem to approach a new standard or resolution of about 16 Megapixel per image

Specialized cameras (e.g., for photogrammetry) are already able to acquire single images of multiples of 100 Megapixels

other methods of generating digital pictures: scanners, biomedical imaging modalities (CT, MRI, and so forth), or computer graphics

for example: by modifying parameters of a flatbed scanner we are able to map an original into images of varying resolution

Early Mathematical Studies

C.F. Gauss and P. Dirichlet (Germany) studied in the first half of the 19th century how the area (contents) of a planar region is related to the number of grid points contained in that area.

C. Jordan (France) initiated then in the second half of the 19th century analogous studies for the contents in 3D space: how the volume can be estimated by the number of grid cubes contained in a 3D set?

Mathematical studies on problems related to grids, or tesselations in 2D or 3D Euclidean space, led to results which prove today to be very useful for picture analysis. For example, a paper by G. Pick (Praha) in 1899 provides a very interesting (accurate) formula for calculating the area of a simple polygon based on contained grid points. Mathematical morphology, initiated in the 19th century by J. Steiner and H. Minkowski (Germany), is another example where basics have been created before computers came into existence, and image analysis benefits now from these fundamental studies.

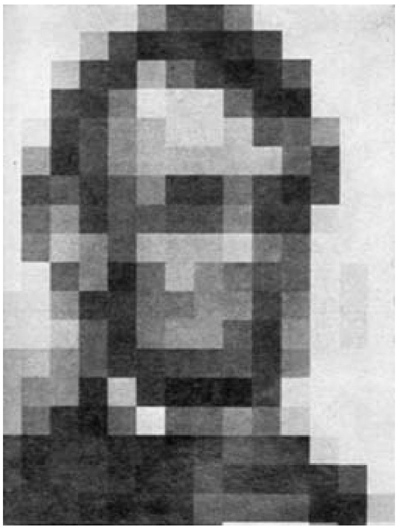
The 1950*s* - 1960*s*

These were the early days of computer-based picture analysis and computer graphics. In those days, a picture might contain only a few thousand pixels.

The “good old times”: a picture was stored in the way of a box of punch cards, for visualization it was common to print letters one on top of the other (for simulating gray levels), and it is reported that some people (in the early 1950 at Carl Zeiss Jena) could even discriminate – by listening to a computer – whether it was multiplying two numbers or adding at that particular moment. Only a few centers worldwide were active in digital imaging, and those were mainly connected to military, medical, space (since it became an issue), or engineering research.

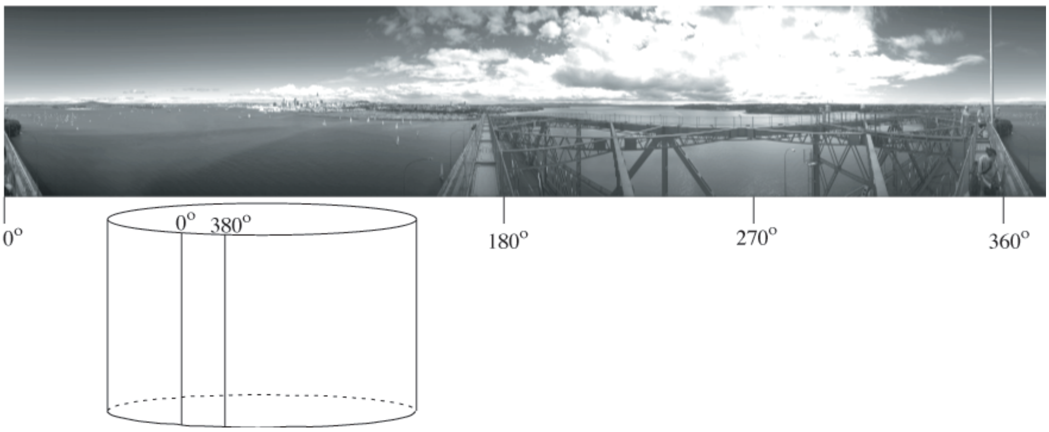
However, in these days already quite basic theoretical work on digital pictures and picture analysis was initiated. For pioneering work, we cite H. Freeman (publications since 1961), J. Bresenham (1963), A. Rosenfeld (1966), U. Montanari (1968), M. Minsky and S. Papert (1969), and J. Sklansky (1970).

Picture Size



Leon Harmon of Bell Labs: picture of Lincoln (252 pixels), "The Recognition of Faces", *Scientific American*, (Nov. 1973).

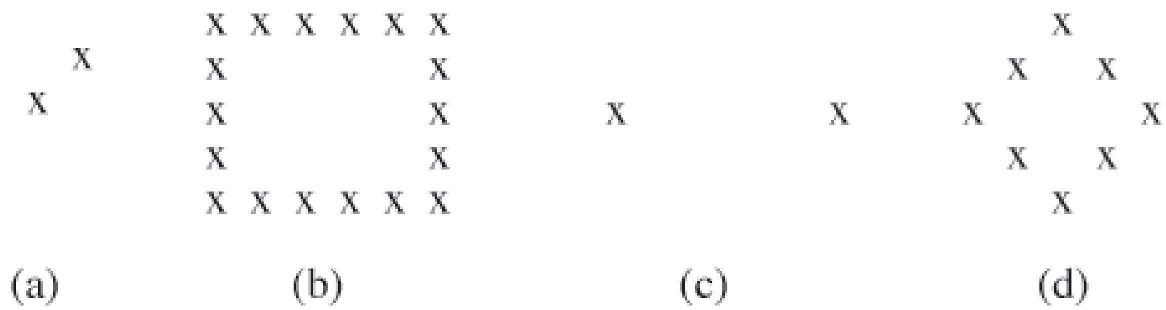
Pictures cannot be arbitrarily large; picture capturing, display, and printing technologies will always impose finite limits.



A 380° panoramic picture of Auckland, New Zealand, captured in 2002 from the top of the harbor bridge. The full-resolution color picture consists of about $10^4 \times 5 \cdot 10^4$ pixels captured on a cylindric surface with a rotating line camera.

4 - and 8-Adjacency

introduced into picture analysis in 1966 by A Rosenfeld and J. Pfaltz: they observed that difficulties arise when 4-adjacency or 8-adjacency (and the corresponding type of *connectedness*) is used for both the 1*s* and 0*s* in a binary picture. The following figure is from this article. The x*s* stand for 1*s* of a binary picture; the 0*s* are not shown.

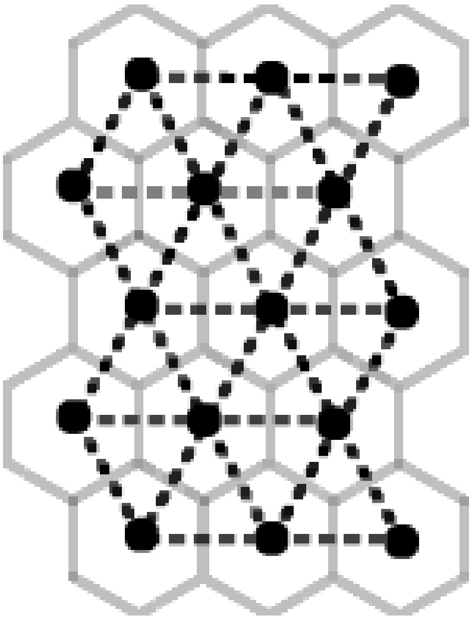


(a) (an object containing two pixels) is 8-connected, as is its complementary set. (b) is both 4- and 8-connected, and its complement is neither 4- nor 8-connected. (c) is neither 4- nor 8-connected, but its complement is both 4- and 8-connected.

> "The 'paradox' of (d) can be (expressed) as follows: If the 'curve' is connected ('gapless') it does not disconnect its interior from its exterior; if it is totally disconnected it *does* disconnect them. This is of course not a mathematic paradox, but it is unsatisfying intuitively; nevertheless, connectivity is still a useful concept. It should be noted that if a digitized picture is defined as an array of hexagonal, rather than square, elements, the paradox disappears."

Hexagonal Array

hexagonal cells, each pixel (i.e., hexagonal cell) has exactly 6 adjacent pixels (not considering special cases at the border of a rectangular picture array)



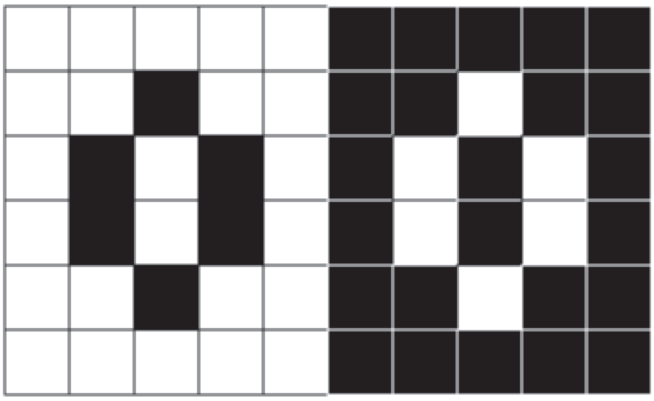
only one type of adjacency: 6-adjacency, and any closed 6-path separates the plane into several 6-components

However: this type of grid is just of theoretical interest - camera and program design decided for the regular orthogonal grid

Connectedness

The reflexive, transitive closure of an adjacency relation on a set M (e.g., of grid points), which is the smallest reflexive, transitive relation on M that contains the given adjacency relation, defines a *connectedness relation*.

M is called *connected* if for all $p, q \in M$ there exists a sequence $p_0, \ldots, p_n$ of elements of M such that $p_0 = p$, $p_n = q$, and p_i is adjacent to $p_{i-1} (1 \leq i \leq n)$; such a sequence is called a *path* and is said to *join* p and q in M.

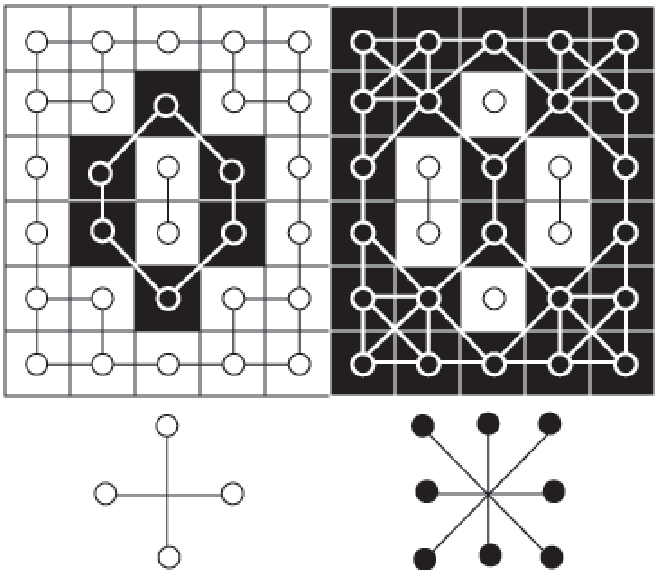


Maximal connected subsets of M are called (connected) *components* of M. What are the 4- or 8-components in the binary picture above?

Components are nonempty and distinct components are disjoint.

Dual Use of 4 - and 8-Adjacency

1967: R.O. Duda, P.E. Hart, and J.H. Munson propose the dual use of 4- and 8-connectedness for 0*s* and 1*s* in a binary picture.



A. Rosenfeld proved in 1974 that this approach allows to avoid conflicts between connectedness and separation as discussed on page 7.

(He showed that the region adjacency graph – to be defined later – of such a binary picture is always a rooted tree.)

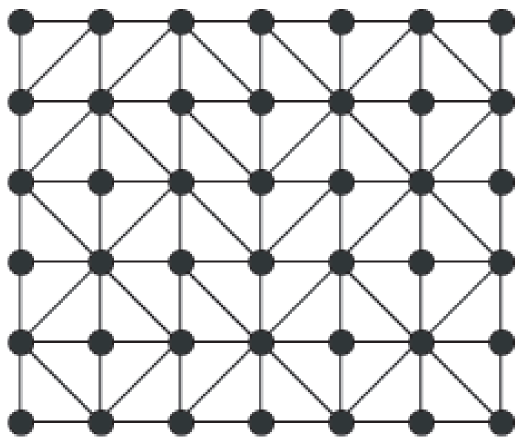
(Another way to solve the problem: all "curves" in the picture should have a width of at least two pixels – as it would be if sampling follows the sampling theorem. Then just 4-adjacency would do. - However, we cannot assume that in general.)

What to Do for Multi-Level Pictures?

Gray-level or color pictures do not allow such a dual use of 4- and 8-adjacency. Here, different methods can be used to ensure a proper duality of connectedness and separation.

Two equivalent options for 2D pictures (to be explained later in detail):

(i) the so-called *switch adjacency*: exactly one "diagonal adjacency relation" in every square defined by four pixels



(ii): consider a "refined grid cell structure" (e.g., including all the 0- or 1-cells in the 2D case) and define a proper adjacency scheme between pixels, for example by using a total order of all possible picture values

Coursework

Related material in textbook: Sections 1.1 and 2.1.3; solve Exercises 1 (page 30) and 10 (page 32).

A.2. [4 marks] Implement a program that does the following:

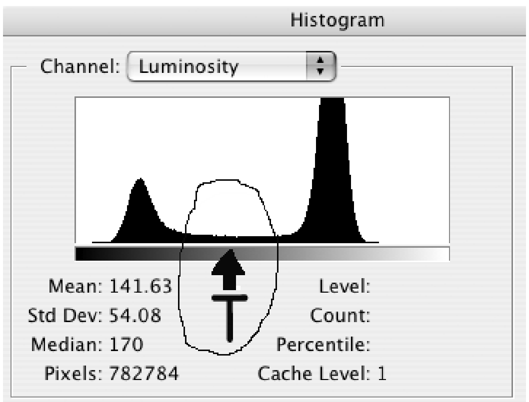
(i) Load and display a gray-level picture P (in *bmp* or *tiff* format) of your choice.

(ii) Display the histogram of your picture P.

(iii) Select a *threshold* T, $0 < T \leq G_{max}$. Map pixel values greater or equal to T onto G_{max}, and all pixel values less than T onto 0. This produces a binary picture where all pixels only have either value 0 (*non-object pixels*) or value G_{max} (*object pixels*).

(iv) Color 4-components of object pixels with different colors.

Repeat the procedure for different thresholds T. The histogram visualized in Step **(ii)** might be helpful for deciding about a threshold. For example, if the histogram of P is *bipolar*



(i.e., two "peaks" separated by one "valley") then it is suggested to take the position of the valley as the threshold for binarization of P.

Appendix: A Few Comments

1. The binary picture on page 9 has 6 white 4-components, but only 2 white 8-components. It also has 6 black 4-components, and 2 black 8-components.

2. For coloring components in **A.2** you can implement either FILL or Rosenfeld-Pfaltz (as discussed in Lecture 3). FILL should be easier to implement, but the latter one might be more interesting. *Note that* FILL *will be useful as a subroutine for later assignments, and its implementation is highly recommended at this stage!*

3. For demonstrating your solution to **A.2** it is fine to work with such input pictures and thresholds where the number of 4-components is reasonably small, say in the range of $10 \cdots 100$. (See comment 3 at the end of Lecture 1, stating the allowed freedom of specifying ”your” way of solving an assignment.)

4. The ”different thresholds” in **A.2** are for demonstrating that your program is able to deal with different input situations. In your submitted solution you may illustrate that for the ”standard case” and also for ”extreme cases”, such as a very low or a very high threshold.

5. Altogether we have 2^{8+8+8} different RGB color values, assuming one Byte for each channel. Of course, not all will be visually distinguishable. For **A.2** it is fine to use a small number (say, 20) of easily distinguishable colors, allowing repetitions if the number of 4-components exceeds 20.

6. An ”object pixel” is either white or black - it is your choice. (In a book, typically we use black object pixels, because the figures are on white paper; on a screen it is preferable to use white object pixels against the dark screen background. In **A.2** you do not have to worry about non-object pixels.