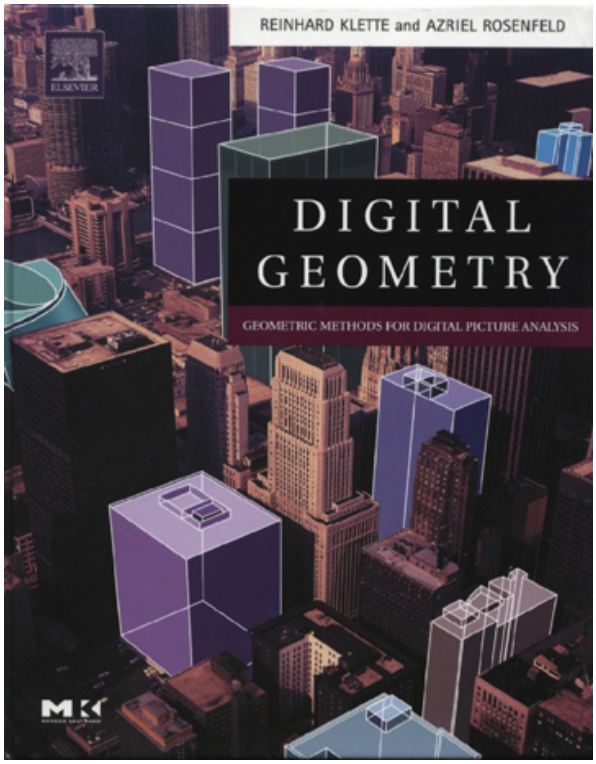
Areas of Application

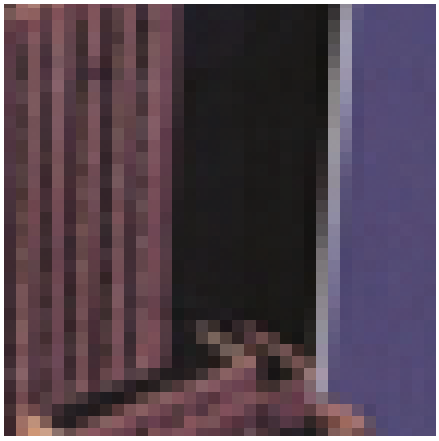
Computer graphics, pattern recognition, image analysis, medical imaging, industrial image analysis, robot vision, and so forth - this lecture discusses fundamentals and basic algorithms for these areas

Textbook

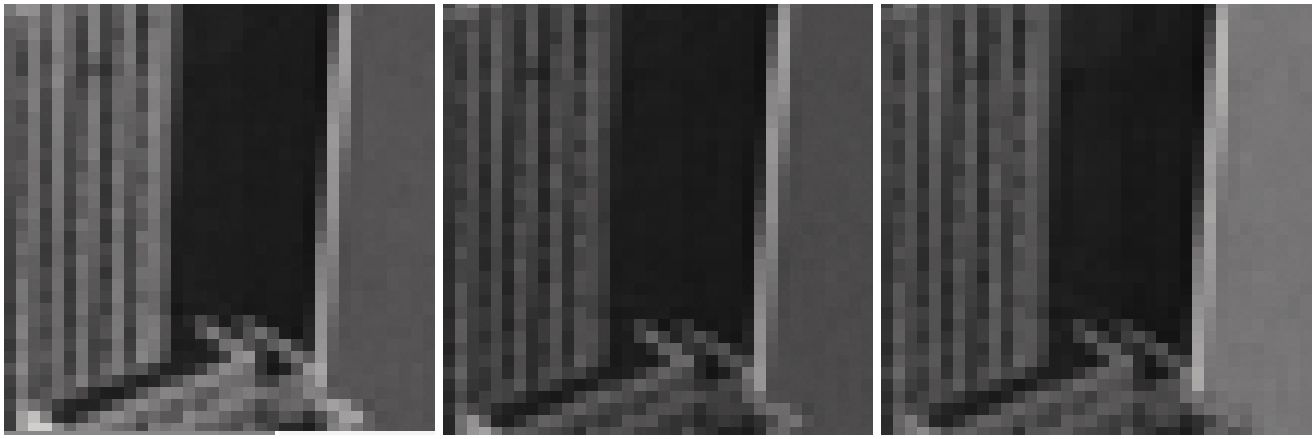


this cover is an example of a

(digital) *color picture* = array of triples (R,G,B) of integers



An RGB picture is composed of three single-valued channels, which can be shown as *gray-level pictures*.

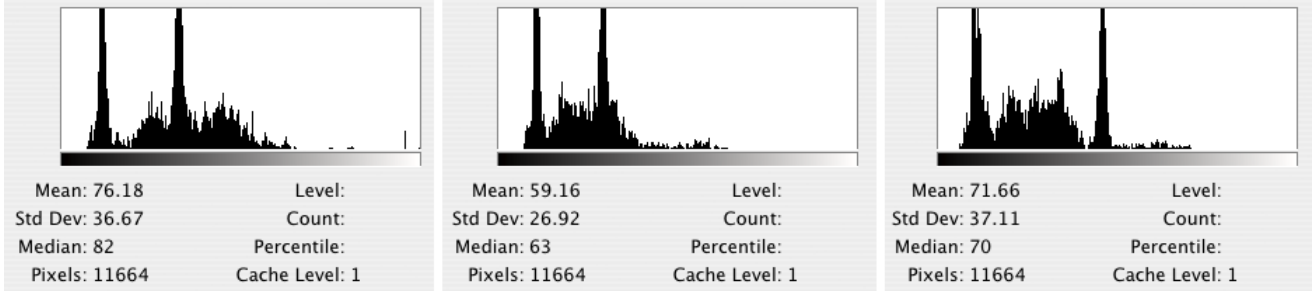


red green blue

gray level = integer between 0 and G_{max}

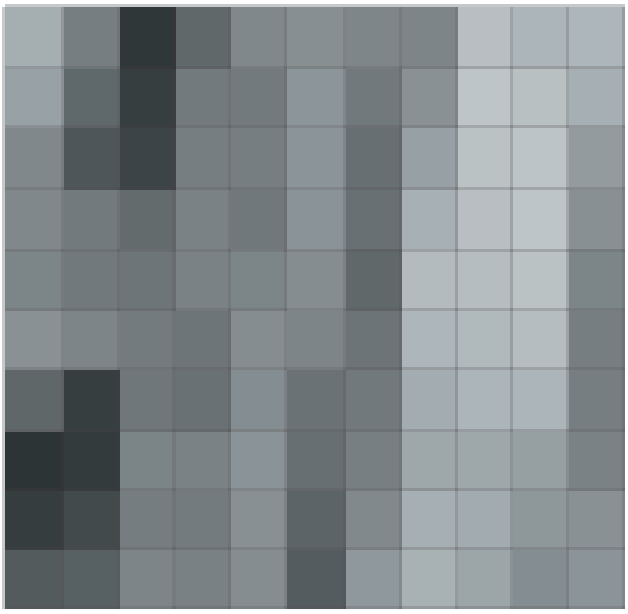
defaults: $G_{max} = 255$ (one byte), 0 = black, G_{max} =white

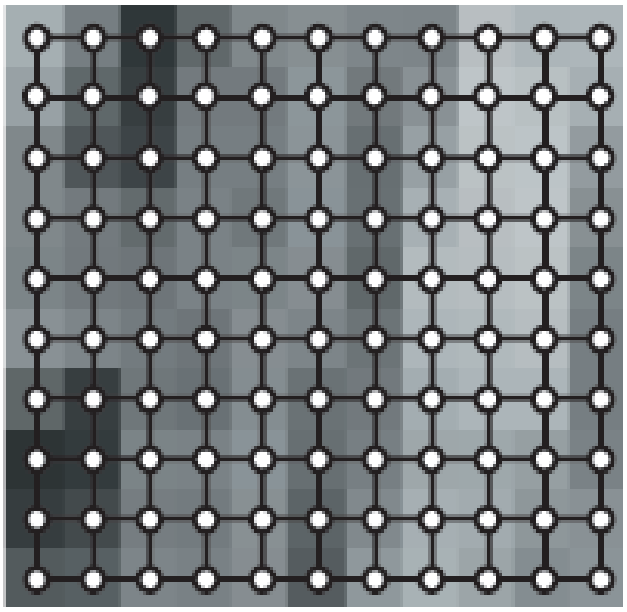
histograms (= frequencies of gray levels) of these three *channels*:



Pictures

pictures have values on a grid of squares or of grid points





left: as on a screen; after zooming in on a picture composed of colored *grid squares*, where the colors (or gray levels) represent values in a (finite) set

right: common array data structure used to represent a picture by values at *grid points*

just two different ways of thinking about pictures in image analysis or computer graphics

binary picture: only two different values, default: values 0 or 1

Pixels and Voxels

A (two-dimensional) 2D digital picture consists of an array of *pixels*, each defined by a location and a value at that location. The term pixel is short for "picture element"; this acronym was introduced in the late 1960*s* by a group at Jet Propulsion Laboratory in Pasadena, California, that was processing pictures taken by space vehicles

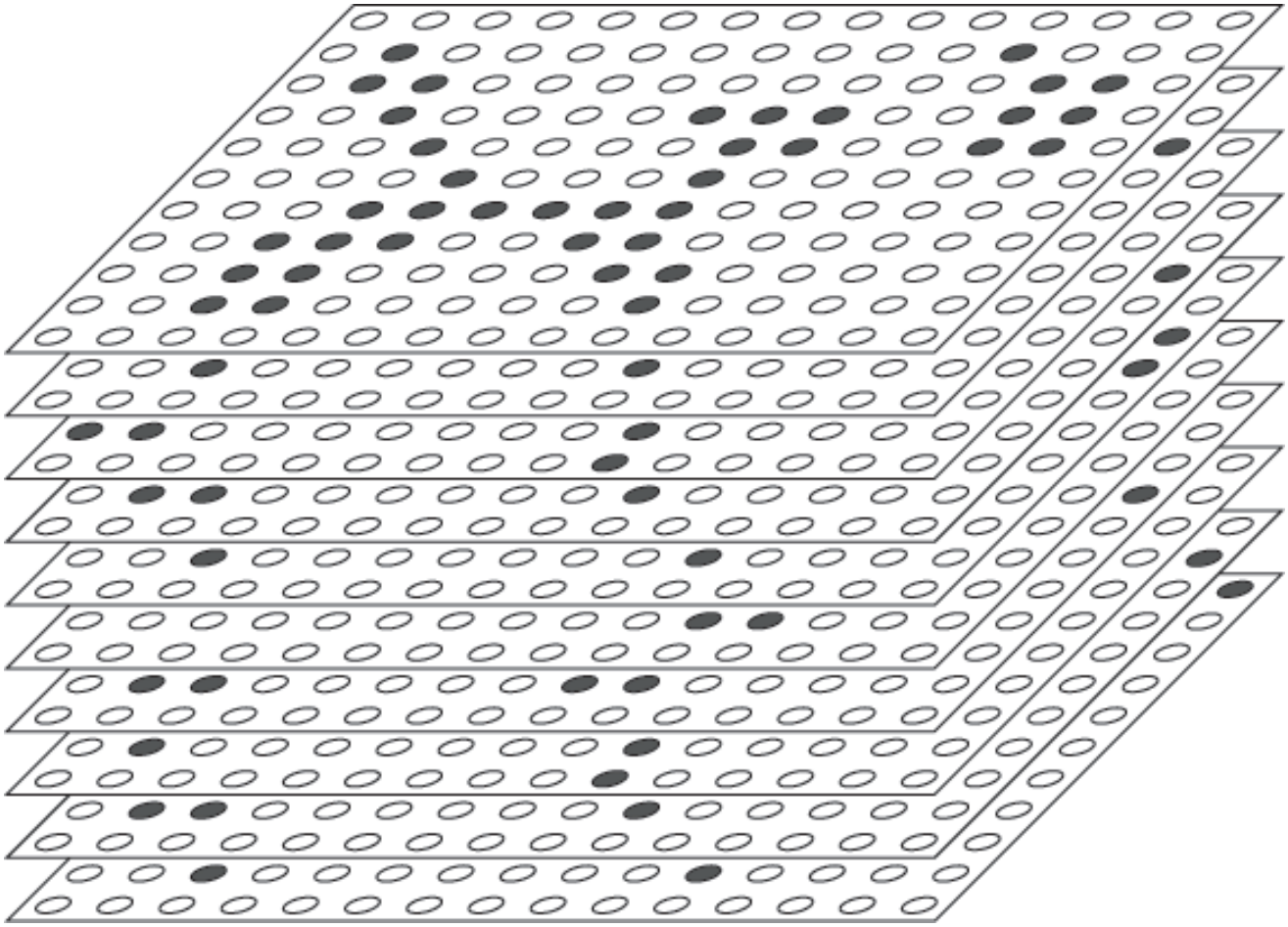
> R.B. Leighton et al. Mariner 6 television pictures: First report. *Science*, **165**:684–690, 1969.

The analogous (three-dimensional) 3D term is *voxel*, which is short for "volume element."

These lectures use the terms *pixel* and *voxel* in a simplified sense; they will (just) refer to the location, which is defined by a regular orthogonal *grid*.

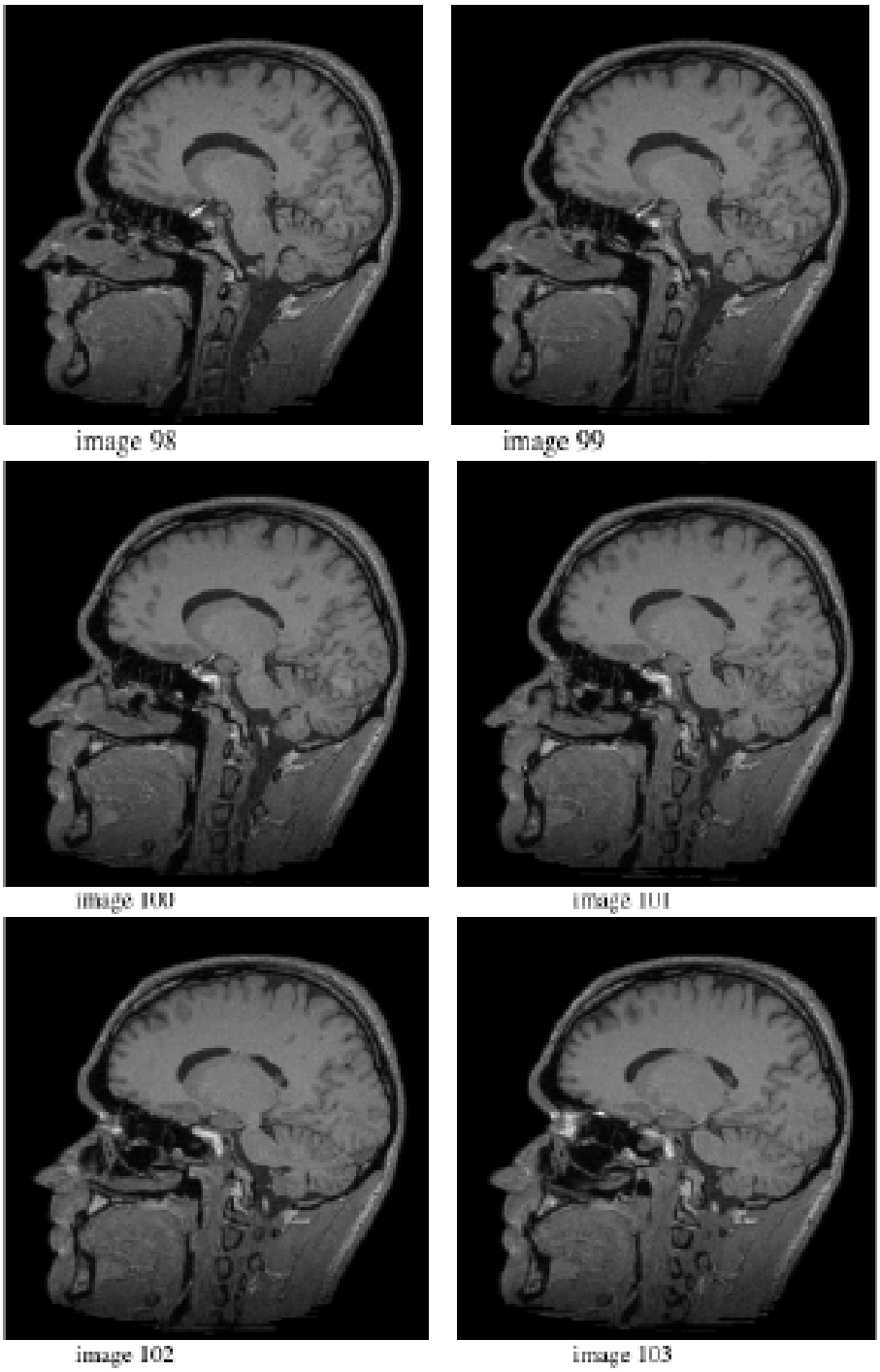
3D Picture in the Grid Point Model

A 3D picture consists of several slices or layers (2D pictures), all of identical size, and alligned within a 3D regular orthogonal grid. The figure shows a binary 3D picture having 10 layers:



A voxel is at a grid point position in such a 3D array (equivalent: a grid cube in the 3D regular orthogonal grid).

six slices of one 3D (magnetic resonance image, MRI) picture



A *picture* is a function defined on the grid that assigns values to pixels or voxels. In general it is produced by some type of imaging process (*an image*).

Regular Orthogonal Grid

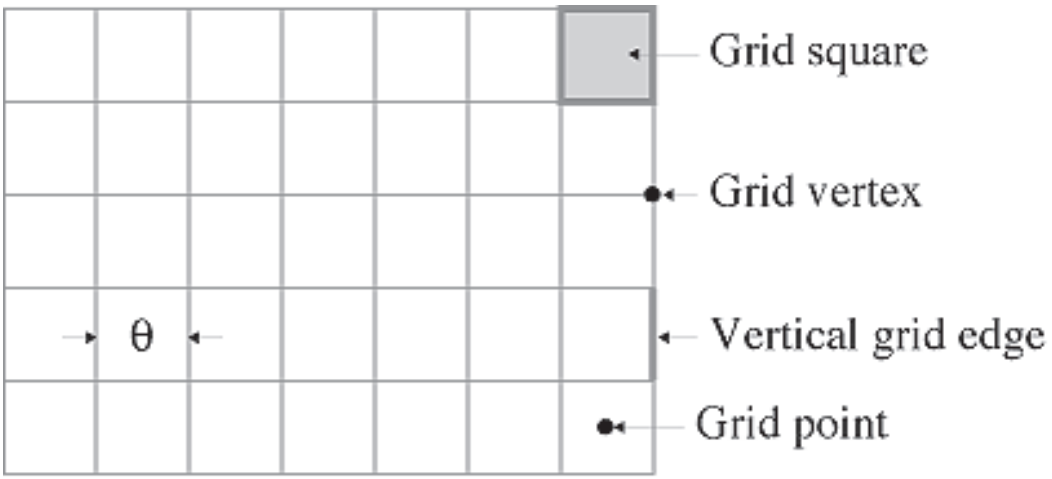
$\mathbb{Z} = \{\ldots, -3, -2, -1, 0, 1, 2, 3, \ldots\}$ = set of integers

$\mathbb{Z}^2 = \{(x, y) : x, y \in \mathbb{Z}\}$ = set of grid points in the plane

$\mathbb{Z}^3 = \{(x, y, z) : x, y, z \in \mathbb{Z}\}$ = set of grid points in the 3D space

grid square: has a grid point in $\mathbb{Z}^2$ as its center point, edges of length 1

grid cube: has a grid point in $\mathbb{Z}^3$ as its center point, edges of length 1



default: length of grid edge $\theta = 1$ (thus: $\mathbb{Z}$ or $\mathbb{Z}^2$ or $\mathbb{Z}^3$ without scaling factor)

A grid cube is also called a *3-cell*; a grid square is a *2-cell*; a grid edge is a *1-cell*; and a grid vertex is a *0-cell* (note: 0,1,2,3 are the dimensions of these cells).

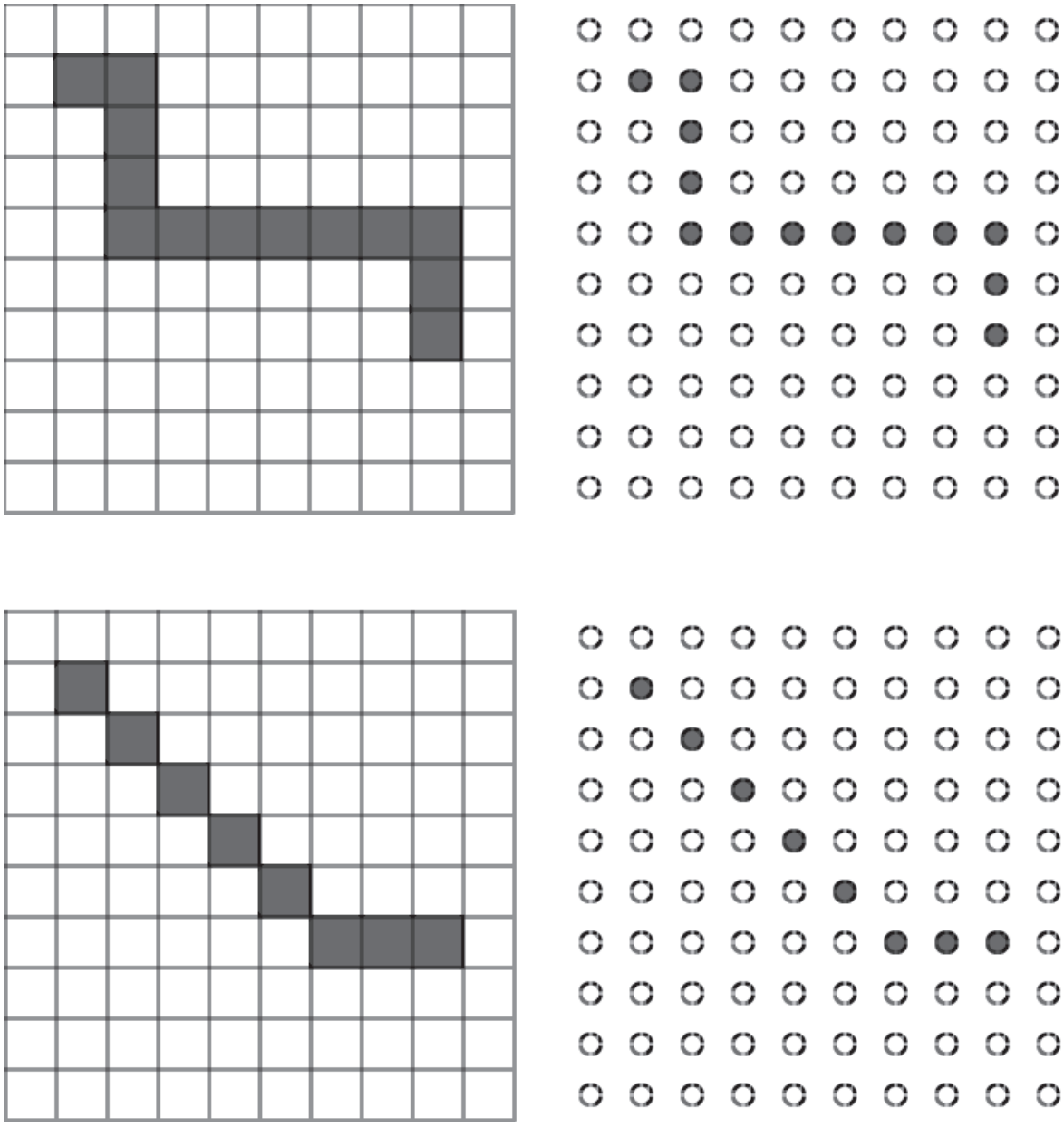
2D Adjacency

Version 1: two 2-cells, c_1 and c_2, are called *1-adjacent* iff $c_1 \neq c_2$ and $c_1 \cap c_2$ is a 1-cell; two grid points $p_1 = (x_1, y_1)$ and $p_2 = (x_2, y_2)$ are called *4-adjacent* iff $|x_1 - x_2| + |y_1 - y_2| = 1$.

Adjacency sets $A_1(c)$ and $A_4(p)$ (left) of one 2-cell c or one grid point p, and the corresponding *neighborhoods* $N_1(c)$ and $N_4(p)$ (right), also containing c or p itself.

Version 2: two 2-cells c_1 and c_2 are called *0-adjacent* iff $c_1 \neq c_2$ and $c_1 \cap c_2$ contains a 0-cell; two grid points $p_1 = (x_1, y_1)$ and $p_2 = (x_2, y_2)$ are called *8-adjacent* iff $\max\{|x_1 - x_2|, |y_1 - y_2|\} = 1$.

Adjacency sets $A_0(c)$ and $A_8(p)$ (left) of one 2-cell c or one grid point p, and the corresponding *neighborhoods* $N_0(c)$ and $N_8(p)$ (right), also containing c or p itself.



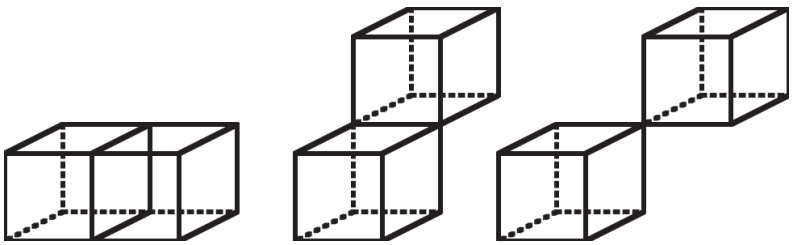
Consecutive sequences of adjacent pixels (also called 'paths') for adjacency relations A_1 and A_4 (top: 1-path and 4-path), A_0 and A_8 (bottom: 0-path and 8-path).

3D Adjacency

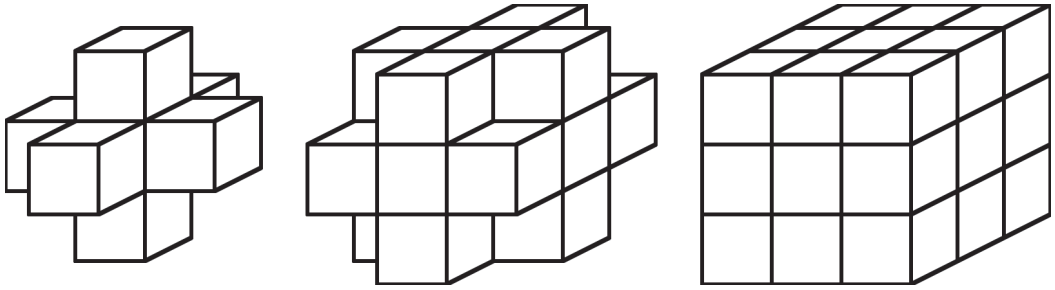
If two points p, q have coordinates $(x_1, \ldots, x_n)$ and $(y_1, \ldots, y_n)$, we define the *Euclidean metric* (*Euclidean distance function*) as follows:

$$d_e(p,q) = \sqrt{(x_1 - y_1)^2 + \ldots + (x_n - y_n)^2}$$

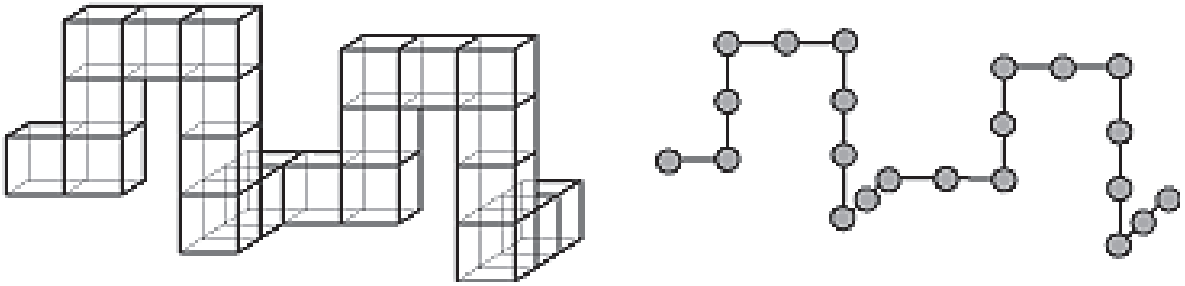
Two 3-cells c_1 and c_2 are called *α-adjacent* iff $c_1 \neq c_2$ and the intersection $c_1 \cap c_2$ contains an α-cell ($\alpha \in \{0, 1, 2\}$). Two 3D grid points $p_1 = (x_1, y_1, z_1)$ and $p_2 = (x_2, y_2, z_2)$ are called *6-adjacent* iff $0 < d_e(p_1, p_2) \leq 1$, *18-adjacent* iff $0 < d_e(p_1, p_2) \leq \sqrt{2}$, and *26-adjacent* iff $0 < d_e(p_1, p_2) \leq \sqrt{3}$.



Left: two α-adjacent 3-cells ($\alpha = 0, 1, 2$). Middle: two α-adjacent 3-cells ($\alpha = 0, 1$). Right: two 0-adjacent 3-cells.



Neighborhoods $N_2(c)$ (left), $N_1(c)$ (middle), and $N_0(c)$ (right) of one 3-cell c.



Consecutive sequences of adjacent voxels for adjacency relations A_2 (left: a 2-path) and A_6 (right: a 6-path).

Coursework

Related material in textbook: Sections 1.1, 2.1.1, 2.1.3, and 2.1.4.

A.1. [3 marks] Implement a program (e.g., in Java, C++, or Matlab) that does the following:

(i) Load and display a color (RGB) picture P (in *bmp* or *tiff* format) of your choice.

(ii) Display the histograms of all three color channels of P.

(iii) Move the mouse cursor within your picture. If it is at pixel p in the picture, compute and display

(iii.1) the outer border of the 11×11 square window $W_{11}(P, p)$ around pixel p in your picture P (i.e., p is the reference point of this window);

(iii.2)(above this window, or in a separate command window) the location p (i.e., coordinates) of the mouse cursor and the RGB values of your picture P at p;

(iii.3) (below this window, or in a separate command window) the intensity value $I(p) = \frac{R(p)+G(p)+B(p)}{3}$ at p; and

(iii.4) MEAN($W_{11}(I, p)$) and VARIANCE($W_{11}(I, p)$) (see definitions of MEAN and VARIANCE in your math lectures; I is the intensity picture).

(iv) Discuss examples of picture windows $W_n(P, p)$ where you see "homogeneous distributions of picture values", and windows showing "inhomogeneous areas". Try to define your definition of "homogeneous" or "inhomogeneous" in terms of histograms, means or variances.

Appendix: A Few Comments

1. Assignment **A.1** is the first of all possible assignments in this lecture. Each lecture ends with specifying one assignment. Each student has to submit a number of assignments as defined by the course coordinator (possibly also based on the specified marks characterizing the complexity of an assignment).

2. Assignment **A.1** is designed to exercise the use of image data and of basic routines for displaying results on a screen. If a student had no prior experience with these subjects then it is highly recommended to work on this assignment. Experienced students may skip this for better challenges later.

3. Assignments **A.1**, **A.2** (end of lecture 2), **A.3** (end of lecture 3), and so forth, are intentionally formulated in a way to offer students a wide range of options for answering them. For example, for assignment **A.1** you can use Java applets to visualize results (but the assignment does not ask for it), you can use small or large sized pictures (the assignment does not specify it), and you can limit cursor movement to a central part of the input picture such that the 11×11 square around location p is always completely contained in your picture (or you can also include special cases when moving the cursor also closer to the picture's border). As a result, every student should come up with her/his individual solution.

4. Web page *citr.auckland.ac.nz/~rklette/Books/MK2004/* contains selected solutions to assignments of this course - not for copying, but for reference or as stimulating (positive) examples (and submissions for publication are welcome).

5. The explaining part of a submitted solution needs to be in pdf format. For example, **(a)** you can write it in MS Word, then go to "print" and save it as a pdf file, or (recommended) **(b)** use one of the Latex editing systems and save the resulting dvi file in pdf format. The size of a pdf file can be (dramatically) reduced by saving it in Adobe Acrobat with the option "reduce file size".

6. No multiple submissions accepted to any of the assignments by the same student - just wait until the deadline is reached and submit only once. Any submitted assignment is marked and contributes to the coursework mark. Multiple submissions or additional submissions (more than as actually required for the coursework) are strictly ignored.

7. **[Outer border]** The outer border in **A.1** is a 13×13 square curve (which could be drawn, e.g., in white) having the recent cursor position at its center. It is expected that you dynamically update this outer border of the 11×11 window when moving the cursor. Alternatively, you could show the 11×11 window also in a second frame on screen. — Following comment 3 above: creative thinking is welcome; a modified solution might be even more elegant than the way suggested in the text. Solutions equivalent in performance (same information to the user, similar run time, etc.) will not lead to any deductions of marks.

8. **[Displaying histograms]** Histograms (as asked for in **A.1**) can be displayed in one frame, or in three different frames. They can be displayed in the same frame as the input picture, or in other frames.

9. **[Gray-level picture]** A gray-level (or gray scale) picture typically has only one channel, with intensities between 0 and 255 (as default, corresponding to one Byte per pixel). However, $R = B = G$ also defines a gray value picture (i.e., we could use the RGB-format of a color picture), but it would be redundant (and a waste of memory, which makes subsequent algorithms more complicated).

10. **[Mean and variance]** The $121 = 11 \times 11$ intensity values in the window define the set of data for which you have to calculate mean and variance.

11. **[Why not jpg format?]** *jpg* is a lossy compression scheme which modifies picture values (between compressed and uncompressed state), and therefor it is not suitable for picture analysis in general. *bmp* or *raw* or *tiff* are formats where pixel values will not be altered by some type of compression mechanism. In *jpg* pictures you can often see a 8×8 block structure due to low-quality compression. – **A.1** says "picture ... of your choice" (not allowing a lossy compression scheme to modify data in some undesired or uncontrolled way), and you can submit your picture with your solution; even better would be that your program is able to work on any picture of specified format and size.

Left: original picture in *tiff* format. Middle: after using *jpg* compression for medium quality. Right: after using *jpg* compression for low quality.