

Digital Image Processing (CPE-415)

Assignment #1 – Solutions

Problems: 2.1, 2.2, 2.5, 2.10, 2.11, 2.12, 2.13, 2.15, 2.16, 2.17

Reference: Gonzalez & Woods, 3rd Edition

Name: _____

Roll No: _____

Section: _____

Due: 27th February 2026 — Marks: 10

Problem 2.1 – Smallest Dot the Eye Can Discern

Given: Fovea = 1.5×1.5 mm, cone density = $150,000/\text{mm}^2$, lens-to-retina = 17 mm, page distance = 0.2 m.

Total cones: $150,000 \times 1.5^2 = 337,500$

Cones per side: $\sqrt{337,500} \approx 581$

One cone diameter: $d_c = 1.5/581 \approx 0.00258$ mm

By similar triangles (object at distance d , image on retina at 17 mm):

$$\frac{x}{200} = \frac{d_c}{17} = \frac{0.00258}{17} \implies \boxed{x \approx 0.0304 \text{ mm} \approx 30.4 \mu\text{m}}$$

The eye cannot see a dot smaller than $\sim 30 \mu\text{m}$ at 20 cm distance.

Problem 2.2 – Dark Theater Adaptation

The process is **brightness adaptation** (dark adaptation).

- Outside: cones active (photopic/bright-light vision), rods bleached.
- Theater: light drops drastically; cones can't function, rods must take over.
- Rods need ~ 20 – 30 min to regenerate rhodopsin \implies the delay before you can see.
- After adaptation: scotopic (rod-based) vision — you see shapes but not colors.

Answer: Brightness adaptation — the eye shifts from photopic (cone) to scotopic (rod) vision, which takes time because rods must regenerate their visual pigments.

Problem 2.5 – CCD Camera Resolution

Given: Chip 7×7 mm, 1024×1024 pixels, distance $d = 500$ mm, focal length $f = 35$ mm.

Scene size (similar triangles): $\frac{\text{Scene}}{500} = \frac{7}{35} \implies \text{Scene} = 100 \text{ mm}$

Pixels/mm on scene: $1024/100 = 10.24 \text{ px/mm}$

Line pairs/mm (Nyquist: 2 pixels per line pair):

$$\boxed{\text{lp/mm} = 10.24/2 = 5.12 \text{ line pairs/mm}}$$

Problem 2.10 – HDTV Movie Storage

Given: 1125 lines, aspect ratio 16:9, 24 bits/pixel (RGB), interlaced (2 fields at 60 fields/s), 2 hours.

- Vertical: 1125 pixels. Horizontal: $1125 \times 16/9 = 2000$ pixels.
- Bits/frame: $2000 \times 1125 \times 24 = 5.4 \times 10^7$
- Frame rate: $60/2 = 30 \text{ fps}$ (interlaced).
- Duration: $2 \times 3600 = 7200 \text{ s} \implies \text{Total frames: } 30 \times 7200 = 216,000$

$$\text{Total} = 5.4 \times 10^7 \times 216,000 = \boxed{1.1664 \times 10^{13} \text{ bits} \approx 1.458 \text{ TB}}$$

Problem 2.11 – Adjacency of S_1 and S_2

Given grid ($V = \{1\}$); S_1, S_2 are the left/right groups of 1-valued pixels.

Key pair: pixel $(4, 5) \in S_1$ and pixel $(4, 6) \in S_2$, both value 1. They are **direct horizontal neighbors** (one step right).

- (a) **4-adjacent: YES.** $(4, 6) \in N_4(4, 5)$.
- (b) **8-adjacent: YES.** 4-adjacency \subset 8-adjacency.
- (c) **m-adjacent: YES.** $(4, 6) \in N_4(4, 5)$ satisfies condition (i) of m-adjacency.

S_1 and S_2 are **adjacent under all three types**.

Problem 2.12 – Converting 8-Path to 4-Path

Algorithm: Scan each step in the 8-path. If a step is 4-connected (horizontal/vertical), keep it. If it is **diagonal** $(x, y) \rightarrow (x \pm 1, y \pm 1)$, replace it with two 4-connected steps:

$$(x, y) \rightarrow (x \pm 1, y) \rightarrow (x \pm 1, y \pm 1)$$

Example: $(0, 0) \rightarrow (1, 1) \rightarrow (2, 1) \rightarrow (3, 2)$

$\Rightarrow (0, 0) \rightarrow (1, 0) \rightarrow (1, 1) \rightarrow (2, 1) \rightarrow (3, 1) \rightarrow (3, 2)$ (bold = inserted pixels)

Each diagonal is decomposed into one horizontal + one vertical move. Either order (horizontal-first or vertical-first) is valid.

Problem 2.13 – Converting m-Path to 4-Path

The algorithm is **identical to Problem 2.12**: replace each diagonal step with two 4-connected steps.

Key difference: In an m-path, a diagonal step from p to q only exists when the shared 4-neighbors $N_4(p) \cap N_4(q)$ have *no* pixels in V . This means the intermediate pixel inserted will **not** have a value in V (it's a background pixel). The resulting 4-path is geometrically valid but may include non- V pixels.

Problem 2.15 – Shortest Paths Between p and q

Grid (coordinates as row, col from top-left):

	0	1	2	3
0	3	1	2	1(q)
1	2	2	0	2
2	1	2	1	1
3	1(p)	0	1	2

(a) $V = \{0, 1\}$

Usable pixels: $(0, 1)=1$, $(0, 3)=1$, $(1, 2)=0$, $(2, 0)=1$, $(2, 2)=1$, $(2, 3)=1$, $(3, 0)=1$, $(3, 1)=0$, $(3, 2)=1$.

4-path: Does not exist. All routes from row 1 \rightarrow row 0 via 4-connectivity are blocked ($(0, 2) = 2$, $(1, 1) = 2$, $(1, 0) = 2$, $(1, 3) = 2$ are all $\notin V$).

8-path: $(3, 0) \rightarrow (3, 1) \rightarrow (2, 2) \rightarrow (1, 2) \rightarrow (0, 3)$. Length = 4. (Step $(3, 1) \rightarrow (2, 2)$: diagonal; step $(1, 2) \rightarrow (0, 3)$: diagonal.)

m-path: The diagonal $(3, 1) \rightarrow (2, 2)$ fails m-adjacency because shared 4-neighbor $(3, 2) = 1 \in V$. Must go through $(3, 2)$:

$(3, 0) \rightarrow (3, 1) \rightarrow (3, 2) \rightarrow (2, 2) \rightarrow (1, 2) \rightarrow (0, 3)$. Length = 5.

(b) $V = \{1, 2\}$

Most pixels usable except $(0, 0)=3$, $(1, 2)=0$, $(3, 1)=0$.

4-path: Manhattan distance = $|3-0| + |0-3| = 6$ (minimum possible).

$(3, 0) \rightarrow (2, 0) \rightarrow (2, 1) \rightarrow (1, 1) \rightarrow (0, 1) \rightarrow (0, 2) \rightarrow (0, 3)$. Length = 6.

8-path: $(3, 0) \rightarrow (2, 1) \rightarrow (1, 1) \rightarrow (0, 2) \rightarrow (0, 3)$. Length = 4.

m-path: Both diagonals $(3, 0) \rightarrow (2, 1)$ and $(1, 1) \rightarrow (0, 2)$ fail m-adjacency (shared 4-neighbors $(2, 0)=1$ and $(0, 1)=1$ are $\in V$ respectively). All useful diagonals blocked \Rightarrow same as 4-path. Length = 6.

Problem 2.16 – D_4 Distance vs. Shortest 4-Path

$D_4(p, q) = |x - s| + |y - t|$ (city-block / Manhattan distance)

(a) D_4 equals the shortest 4-path length when the path is **unconstrained** — i.e., V contains all pixel values in the image (no obstacles). Any blocking pixel forces a detour, making the 4-path longer than D_4 .

(b) **Not unique.** The path requires $|x-s|$ horizontal and $|y-t|$ vertical steps, which can be interleaved in $\binom{|x-s|+|y-t|}{|x-s|}$ ways. Unique only if p, q share a row or column.

Problem 2.17 – D_8 Distance vs. Shortest 8-Path

$D_8(p, q) = \max(|x - s|, |y - t|)$ (chessboard distance)

(a) D_8 equals the shortest 8-path length when **all pixels are in V** (no obstacles) — same condition as 2.16 but for 8-paths. Strategy: make $\min(|x-s|, |y-t|)$ diagonal moves then $||x-s| - |y-t||$ straight moves = $\max(|x-s|, |y-t|)$ total.

(b) **Not unique.** Diagonal and straight moves can be interleaved in different orders. Unique only if distance is 0 or 1, or p, q share a row/column/diagonal with distance 1.