ZENITH STUDY HUB

Artificial Intelligence for Robotics

UNIT V | MACHINE VISION

Digital images, centroids, filtering, edge detection, segmentation, matching and compression

A syllabus-complete, worked and illustrated learning edition

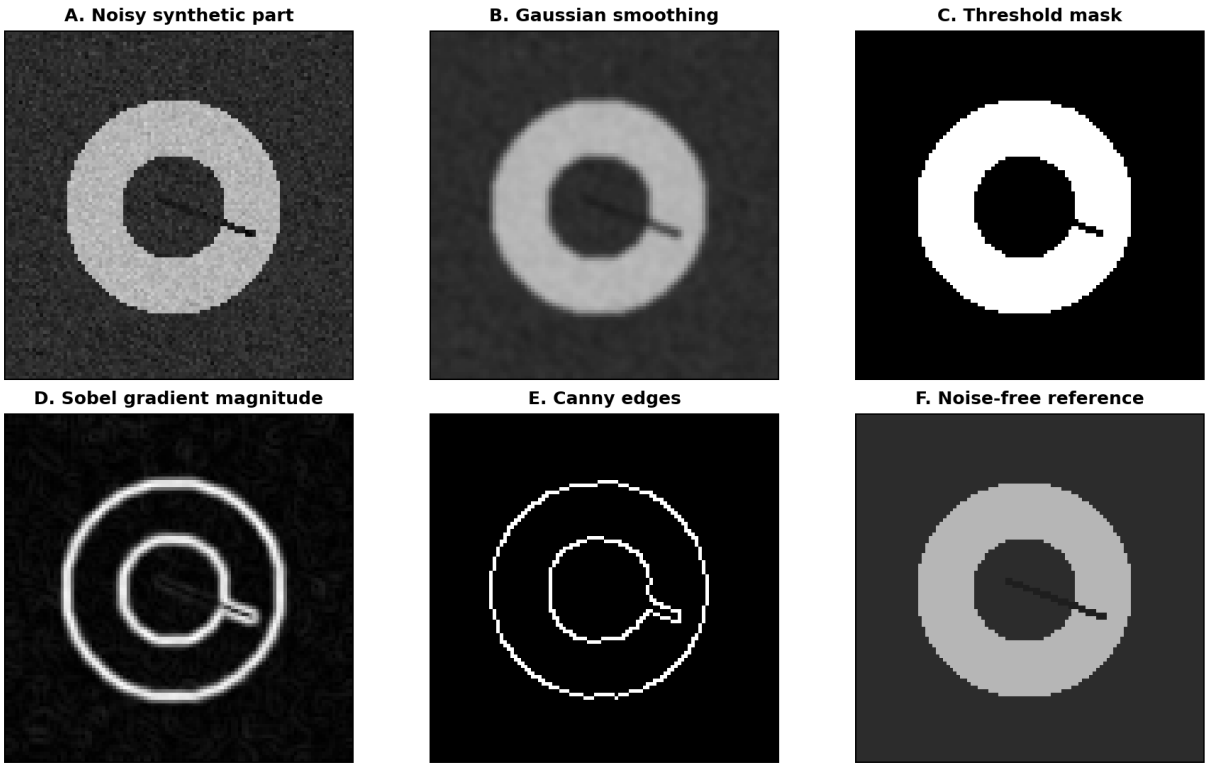

Figure 1. Original synthetic manufactured-part image, shown through smoothing, thresholding, gradient computation and Canny detection. Each panel is generated from an explicitly defined simulated array; none is a real factory photograph.

Course: PCC301-ROA, SPPU TE Robotics & Automation, 2024 Pattern.

Prepared for: Om Deshmukh | 21 November 2026 | Academic review candidate

Your learning map: syllabus to practical skills

Syllabus map verified
This unit follows the exact topic list on supplied *TE Syllabus.pdf*, physical PDF page 15. Each method is treated as a learning concept, not as a hardware-certified robot design.

Syllabus block	What you will be able to calculate or explain
Digital image types / resolution	Represent binary, grayscale and RGB pixels; estimate memory use.
Centroids / moments	Calculate binary, weighted-gray and explicitly weighted RGB centres.
Processing stages	Select illumination, preprocessing, segmentation and decision steps.
Smoothing / sharpening	Apply a 3×3 mean or high-pass kernel correctly.
Threshold / gradients	Apply a threshold and evaluate gradient response.
Sobel / Prewitt / Roberts / LoG / Canny	Sketch kernels, explain their benefits and limitations.
Four segmentation methods	Distinguish threshold, region, edge and K-Means grouping.
SSD and NCC	Compute matching scores and know which direction is better.
Lossless compression	Trace RLE, Huffman and LZW without losing symbols.
Lossy / JPEG / DCT / Wavelet	Explain transform, quantization and entropy coding.
Machine vision systems	Design a conveyor-sorting and robot-guidance pipeline.

In simple language
Machine vision lets a robot use measurements made from images. An image is not yet a robot command: we must illuminate the object, obtain useful pixels, locate the target, convert image coordinates into calibrated physical coordinates, check confidence and only then issue a permitted action.

Safety and source note. Every example image and plotted kernel in this PDF is an *original synthetic teaching example*, not a camera measurement from an industrial inspection station. Practical robotic implementations need calibration and independent protective controls. Images retrieved on the web were researched for suitability, but no third-party photograph is embedded because the source files could not be transferred reliably into this build.

1 Digital Images and Resolution

1.1 What is a digital image?

A camera converts light falling on a sensor into measured values at a regular grid of picture elements, called **pixels**. A digital image is commonly represented as an array $I(y, x)$, where x counts columns to the right and y counts rows downward, starting with $(x, y) = (0, 0)$ at the top-left corner. We will use this convention throughout.

In simple language
Think of an image as a spreadsheet: each cell corresponds to a tiny area of the scene. A binary cell can say whether an object is present; a gray cell says how bright that area appears; three colour numbers describe red, green and blue components.

1.2 Binary, grayscale and RGB

Binary: one bit per ideal pixel, often 0 for background and 1 for foreground. In many software APIs masks actually use 8-bit storage with values 0 and 255; storage depends on the format and implementation.

Grayscale: one scalar intensity per pixel, commonly 0–255 for an 8-bit image.

RGB: three channels per pixel, red, green and blue. In an 8-bit-per-channel image, each channel is usually 0–255; a colour image therefore uses 24 raw bits per pixel before compression.

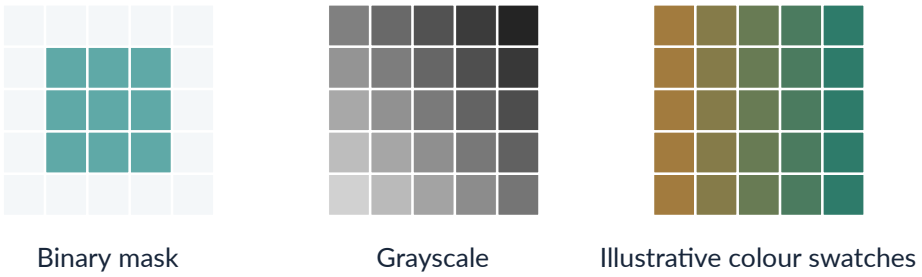

Figure 2. Original aligned 5×5 pixel grids. The colour panel is a schematic, not an encoded sRGB measurement. Binary and grayscale examples represent different possible storage models.

1.3 Resolution, sampling and raw memory

Image resolution is the number of pixel samples along each axis, for example 1920×1080. Higher resolution can help measure small details, but it does not fix blur, poor optics, weak contrast or inaccurate calibration. The actual physical sampling scale depends on the camera field of view (FOV), perspective and lens distortion.

Worked example: pixel count and memory
An uncompressed RGB image has 640×480 pixels, with 8 bits in each of three channels. Pixel count $= 307{,}200$. Raw memory $= 307{,}200 \times 3 = 921{,}600$ bytes $= 0.9216$ MB (decimal), or 0.8789 MiB. This excludes row padding and metadata.

Ideal scale example
A perpendicular flat surface spans 200 mm across a 1000-pixel image width. Nominal scale $s = 200/1000 = 0.2$ mm/pixel. A 20-pixel displacement corresponds to 4 mm *only* if the camera is calibrated and perspective/height changes are negligible.

2 Centroid and Image Moments

2.1 Why a robot needs the centroid

A **centroid** is the average location of an object's pixels under a stated weighting rule. A robot may use it to estimate the location of a detected part on a belt. Its orientation, grasp point, surface normal and safety clearance must be computed separately: centroid alone is not a collision-free grasp pose.

2.2 Binary centroid (unweighted)

For N foreground pixels with coordinates (x_i, y_i),

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i, \qquad \bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i.$$

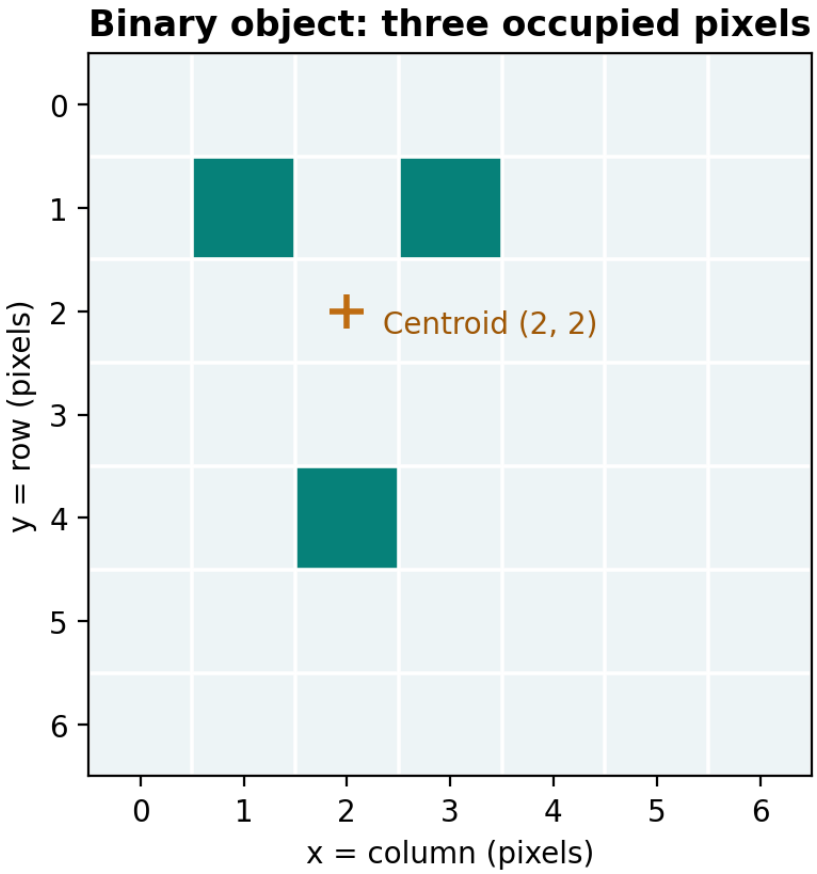

Figure 3. Binary-object centre at $(\bar{x}, \bar{y}) = (2, 2)$ for foreground pixels $(1, 1)$, $(3, 1)$ and $(2, 4)$. The image uses zero-based columns x and rows y increasing downward.

Binary-mask numerical
Pixels: $(1,1),(3,1),(2,4)$. Then $N=3$; $\bar{x}=(1+3+2)/3=2$; $\bar{y}=(1+1+4)/3=2$. Verify the centroid is *not necessarily itself a foreground pixel*.

2.3 Grayscale intensity-weighted centroid

Let nonnegative grayscale weights be $w_i = I(y_i, x_i)$ after a chosen background subtraction. Define spatial moments

$$m_{00}=\sum_i w_i, \quad m_{10}=\sum_i x_i w_i, \quad m_{01}=\sum_i y_i w_i; \quad (\bar{x},\bar{y})=\left(\frac{m_{10}}{m_{00}},\frac{m_{01}}{m_{00}}\right).$$

Grayscale numerical
Three pixels have $(x,w)=(0,1),(1,2),(2,3)$, all at row $y=1$. Thus $m_{00}=6$, $m_{10}=0+2+6=8$, $m_{01}=6$, so $(\bar{x},\bar{y})=(4/3,1)$. The brighter pixel at $x=2$ pulls the result to the right.

2.4 RGB centroid: state the weights first

An RGB triplet alone does *not* define a unique weighted centroid. Possible choices are a per-channel centroid or a centroid of a thresholded colour mask. For an illustrative *channel-sum* weighting rule, let $w_i = R_i + G_i + B_i$ (not calibrated physical luminance). For two pixels at $x=0,2$, with sums $60, 180$, we obtain $\bar{x}=(0\cdot 60+2\cdot 180)/240=1.5$. A perceptual-luminance choice requires correctly handling colour space and gamma; always document it.

Undefined case: $m_{00}=0$ for an empty binary mask or zero total weight. No centroid exists. Do not silently report $(0,0)$; mark detection failed.

3 The Image Processing System

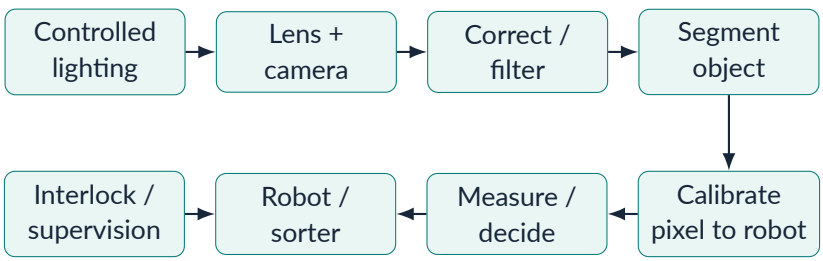

Figure 4. Original industrial vision flow. Protective interlocks are separated from the machine-vision inference chain; calibration is required before robot motion.

Acquisition: choose illumination (backlight, coaxial, ring light), camera exposure, shutter, focus and lens. Backlighting gives silhouettes; front lighting exposes texture.

Preprocessing: denoise, adjust contrast if justified, correct lens distortion and normalize illumination.

Analysis: threshold or segment the target, compute object features, perform template/shape recognition and estimate measurement uncertainty.

Output: map pixels to physical coordinates using calibration, pass the measurement through a decision gate, then command the controller.

Practical risk. A 20 ms image pipeline plus 30 ms communication and 70 ms robot response implies at least 120 ms response latency, ignoring exposure timing and scheduling jitter. A fast conveyor may move substantially during that delay. Use time stamps and position compensation.

4 Smoothing, Noise and Sharpening

4.1 Mean and Gaussian smoothing

A spatial filter computes a new pixel from neighbouring values. A normalized mean filter is

$$K_{\text{mean}} = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

A Gaussian-like normalized 3×3 filter is

$$K_G = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}.$$

Smoothing reduces random high-frequency noise but can blur thin edges and scratches. A larger kernel does not always mean a better inspection system: the defect itself may disappear.

3×3 mean-filter numerical

For patch $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 90 & 0 \\ 0 & 0 & 0 \end{bmatrix}$, mean-filter output at the centre is $(90)/9 = 10$. The bright impulse becomes less extreme, but is spread across nearby outputs. This is a *centre-output* calculation; the border rule is irrelevant at this interior pixel.

4.2 Sharpening and unsharp masking

A simple sharpening kernel is

$$K_{\text{sharp}} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}.$$

On a constant patch of intensity 100, the centre output is $5(100) - 4(100) = 100$, as expected: a constant region remains constant. On a patch with centre 130 and four direct neighbours of 100, it produces $5(130) - 4(100) = 250$, emphasizing the local difference. Values outside the permitted data range must be represented in higher precision or appropriately clipped.

Unsharp masking computes $I_{\text{sharp}} = I + \alpha(I - I_{\text{blur}})$. It can reveal fine features, but also amplify noise or halos.

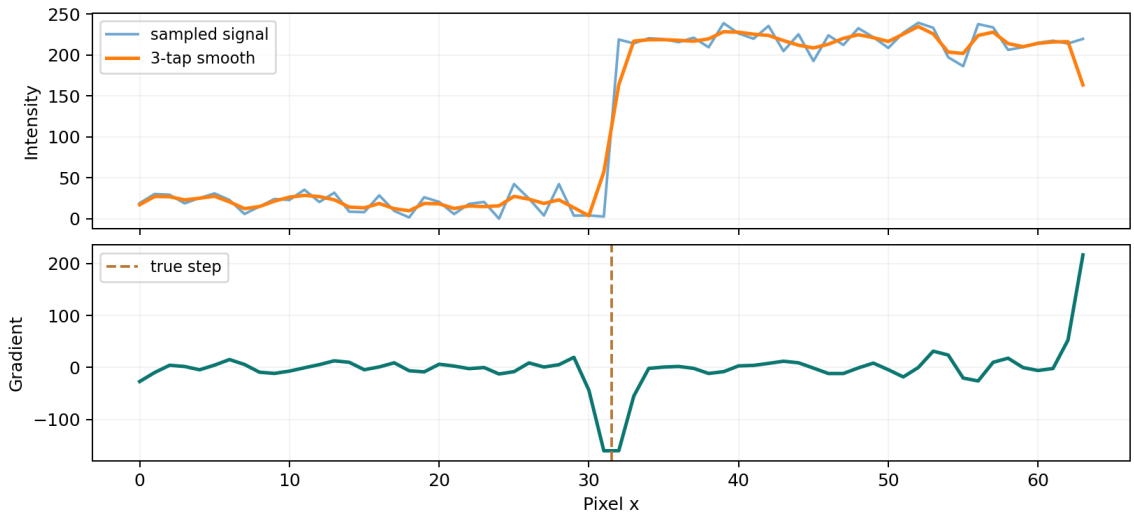

Figure 5. Original synthetic intensity step and discrete gradient response. The highlighted boundary illustrates why smoothing and differentiation should be chosen together.

5 Thresholding and Image Gradients

5.1 Global versus adaptive thresholding

A threshold creates a mask by a declared rule; for example $B(x, y) = 1$ when $I(x, y) > T$, and $B = 0$ otherwise. With $T = 100$, intensities $35, 120, 180$ produce $0, 1, 1$. Pixel exactly equal to 100 is background under the strict greater-than convention.

Global thresholding uses one T everywhere and works when foreground/background intensities separate consistently. **Adaptive thresholding** varies T locally to handle lighting gradients. **Otsu's method** chooses a threshold based on the image histogram under a criterion of class separation; it is not guaranteed to work for every illumination pattern.

5.2 What a gradient means

The image gradient $\nabla I = (G_x, G_y)$ approximates the local intensity change horizontally and vertically. Edge strength is often $\sqrt{G_x^2 + G_y^2}$; a less expensive approximation is $|G_x| + |G_y|$. State which one is used. Orientation can be computed with atan2(G_y, G_x).

Gradient magnitude
If $G_x = 3$ and $G_y = 4$, Euclidean edge strength is $\sqrt{3^2 + 4^2} = 5$ intensity units per chosen sampling step. The L1 approximation would be $3 + 4 = 7$, not the same result.

6 Sobel, Prewitt, Roberts, LoG and Canny

6.1 Five methods in simple language

Sobel: smooths modestly along one direction and estimates derivative along the perpendicular direction.

Prewitt: similar 3×3 derivative but with equal transverse weights.

Roberts cross: uses a tiny 2×2 diagonal difference; very local, often more noise-sensitive.

Laplacian of Gaussian (LoG): smooths then applies a second derivative; zero crossings can indicate boundaries, but multiple responses and noise matter.

Canny: a multi-stage detector: smoothing, gradient computation, non-maximum suppression and dual-threshold hysteresis linking. It is not merely a single convolution mask.

6.2 Specify kernel signs explicitly

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \qquad P_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \qquad R = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

These masks show one common sign convention; reversing the mask sign reverses derivative sign but preserves gradient magnitude. The specific Roberts kernel is one diagonal component; its second orthogonal component must also be computed for full gradient magnitude.

Sobel 3×3 derivative by hand
For a vertical brightness ramp

$$I = \begin{bmatrix} 10 & 20 & 30 \\ 10 & 20 & 30 \\ 10 & 20 & 30 \end{bmatrix}, \quad S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix},$$

pointwise kernel sum (correlation convention) gives $G_x = (-10 + 30) + (-20 + 60) + (-10 + 30) = 80$, while the corresponding S_y response is zero. If the filter operation uses mathematical convolution with a flipped kernel, sign reverses; state your convention when interpreting signs.

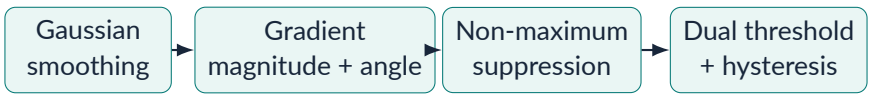

Figure 6. Original Canny processing stages. Hysteresis keeps weak edge pixels only when appropriately connected to stronger detections under the chosen connectivity rule.

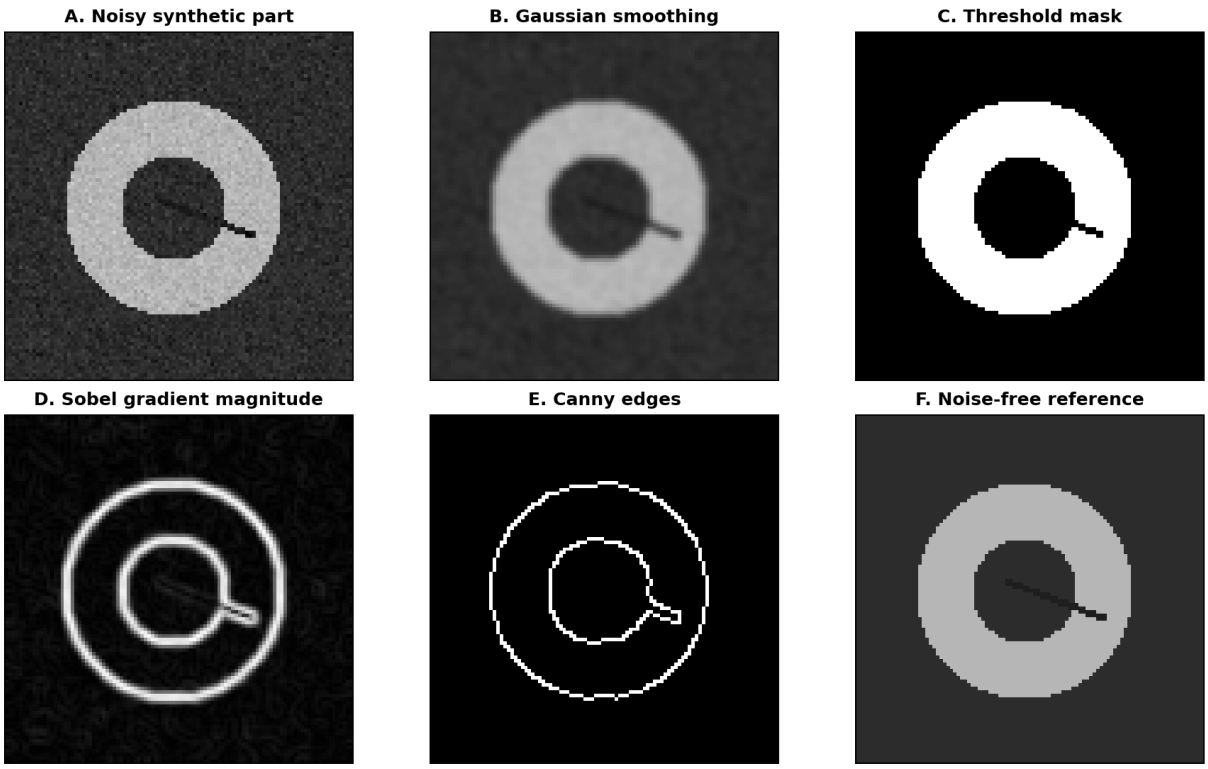

Figure 7. Original synthetic part under Gaussian smoothing, thresholding, Sobel-gradient magnitude and Canny. Thresholding creates a region mask; Canny represents boundaries. Both outputs depend on algorithm parameters.

Detector comparison trap. Sobel, Prewitt, Roberts and LoG have different noise and scale behaviour. Do not say that one algorithm is always best. Tune using measured defect sizes, known lighting, computational budget and false-alarm tolerance.

7 Four Segmentation Methods

7.1 Threshold-based, region-based and edge-based

Segmentation partitions the image into meaningful regions or class masks.

Threshold-based: split by intensity or colour ranges. Fast, but shadows and touching objects can merge.

Region-based: start from a seed and add neighbouring pixels that meet a similarity rule. Sensitive to seed choice and stopping criterion.

Edge-based: detect boundaries and then close/interpret them as regions.

Open or weak contours may produce incomplete regions.

K-Means: treat pixels as feature vectors (e.g. intensity or RGB), assign to nearest cluster centre, then update centre means; spatial smoothness and object identity are not automatically enforced.

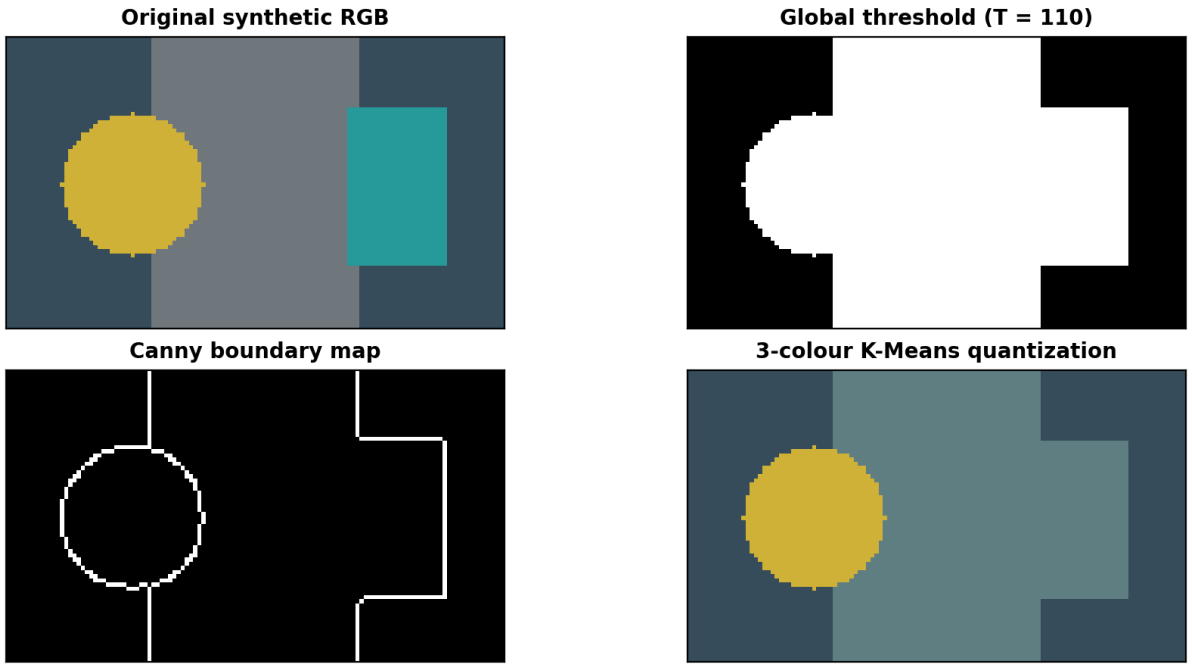

Figure 8. Four views of one original synthetic scene. The K-Means panel quantizes RGB colours into three clusters; its labels are *not* automatically the physical-object classes.

K-Means segmentation step
For grayscale intensities $[10, 20, 200, 220]$ and initial centres $(15, 210)$, nearest-centre assignments are $\{10, 20\}$ and $\{200, 220\}$. Updated centres are $(15, 210)$, so this tiny example is already stable. In a real image with similar backgrounds and objects, colour-only clustering can fail.

7.2 Connected components and bounding boxes

After creating a binary foreground mask, connected-component labelling groups neighbouring foreground pixels. A rectangular bounding box gives approximate location and size; its centre differs from the shape centroid for asymmetric objects. Four-connectivity and eight-connectivity can produce different component counts at diagonal contacts.

In simple language
A segmentation mask says *which pixels belong to a selected region*. It does not, by itself, tell the robot whether the region is a bolt, a reflection, a defect or an occluded object.

8 Image Matching: SSD and NCC

8.1 Sum of squared differences (SSD)

A template is a small image patch slid over the search image. For candidate patch I and reference template T with identical pixel indexing,

$$\mathrm{SSD}(I,T)=\sum_i (I_i-T_i)^2.$$

Smaller SSD means a closer raw-intensity match, with exact zero for identical values. SSD is sensitive to brightness changes.

SSD numerical
$T=(2,4,6)$ and $I=(1,5,7)$. SSD $=(1-2)^2+(5-4)^2+(7-6)^2=3$. A competing patch $(2,4,9)$ has SSD $=9$, so I matches more closely by SSD.

8.2 Zero-mean normalized cross-correlation (ZNCC)

For vectors with nonzero variance,

$$\mathrm{ZNCC}(I,T)=\frac{\sum_i (I_i-\bar{I})(T_i-\bar{T})}{\sqrt{\sum_i (I_i-\bar{I})^2 \sum_i (T_i-\bar{T})^2}}.$$

This score ranges between -1 and $+1$ for conventional real-valued vectors: $+1$ represents a perfectly aligned linear pattern, -1 an inverted one. **Higher is better** for ordinary positive-template matching. If either patch is constant, the denominator is zero and the score is undefined by this formula.

NCC numerical
For $T=(1,2,3)$ and $I=(2,4,6)$, both centred patterns point in the same direction: the score is exactly +1. Scaling intensities by a positive constant preserves ZNCC, but blur and geometric distortions can change the score.

8.3 Matching workflow and common traps

Decide the search region, allow permitted translations/rotations/scales, calculate the score map, choose the best candidate, then reject if the match is weak or inconsistent with expected geometry. Do not assume a high score proves object identity under changing appearance.

Exam trap: SSD should be *minimized*; ordinary ZNCC should be *maximized*. A zero-variance window requires explicit handling rather than division by zero.

9 Lossless Image Compression

9.1 Why compress images?

Saving or transmitting raw vision images can consume storage and communication bandwidth. Compression replaces redundant representation with a shorter encoding. In **lossless** compression, decoding reconstructs the exact original symbols; in **lossy** compression, some information may be irreversibly discarded.

9.2 Run-length encoding (RLE)

RLE stores value–run-length pairs. For binary row $[0, 0, 0, 1, 1, 0]$, a possible encoding is $(0, 3), (1, 2), (0, 1)$, provided the decoder knows the representation and length. Alternating symbols can make RLE larger, not smaller.

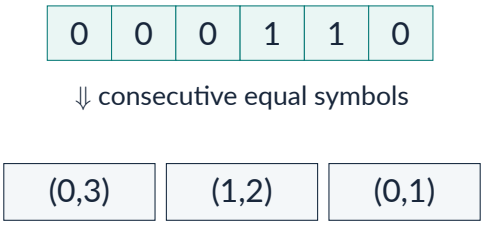

Figure 9. Original RLE demonstration. This example stores three pairs rather than six individual values; total compression depends on the byte-level format chosen for each pair.

9.3 Huffman coding: short codes for frequent symbols

Huffman coding builds a prefix-free binary code tree using symbol frequencies. No valid codeword is a prefix of another. For counts $A = 4, B = 2, C = 1, D = 1$, combine the two smallest counts (1+1=2), combine two 2s (=4), and combine the two 4s (=8). One valid resulting code family is $A = 0$, $B = 10$, $C = 110$, $D = 111$. Mean bits per symbol are $(4 \cdot 1 + 2 \cdot 2 + 1 \cdot 3 + 1 \cdot 3)/8 = 14/8 = 1.75$ bits, before tree-description overhead.

9.4 LZW: build a dictionary of repeated sequences

Lempel–Ziv–Welch (LZW) starts with a dictionary of elementary symbols and adds longer phrases as repeated patterns appear. Encoder and decoder reproduce the same dictionary growth rules, permitting exact reconstruction. Unlike static Huffman, LZW learns phrases rather than only assigning variable-length codes to single symbols. It may be ineffective on random data; format overhead matters.

Compression warning. Do not claim every lossless algorithm compresses every image. Entropy and file-format overhead put practical limits on achieved compression.

10 JPEG, DCT and Wavelet Coding

10.1 The baseline JPEG pipeline

In a common baseline JPEG workflow, image colour samples are optionally transformed into a luminance/chrominance representation; the image

is divided into 8×8 blocks; values are level-shifted; a **discrete cosine transform** (DCT) is applied; DCT coefficients are **quantized**; then reordered and entropy-coded. Quantization is usually the dominant lossy step. DCT itself is a mathematical transform, not automatically a lossy encoder.

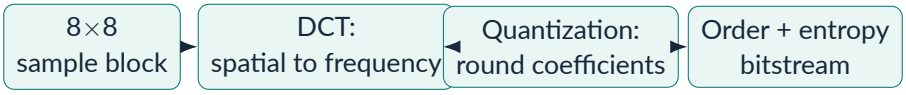

Figure 10. Original simplified baseline JPEG image-coding chain. Actual codecs include syntax markers, colour handling and other details; compression is not just DCT alone.

10.2 Why JPEG introduces artifacts

Strong quantization discards fine variations in the frequency coefficients and can create blocking or ringing near sharp boundaries. A machine-vision defect occupying only a few pixels may disappear after compression, so compare detection on original and compressed images before accepting a recording format.

Illustrative quantization arithmetic
A transform coefficient $c = 23$ with a chosen positive quantization step $q = 10$ is represented approximately by $\hat{c} = \text{round}(23/10) = 2$. On reconstruction, $\widetilde{c} = 2 \cdot 10 = 20$, losing 3 coefficient units. This demonstrates quantization; it is not an example of a full JPEG file.

10.3 Wavelet transform and JPEG 2000 distinction

A wavelet transform represents image structure at multiple spatial scales, such as coarse approximation plus horizontal/vertical/diagonal detail bands. Some wavelet-based coding systems, including JPEG 2000, may be used in lossy or lossless modes depending on transform and coding choices. **Ordinary baseline JPEG is DCT-based, not wavelet-based.**

In simple language
Lossless: reconstruct every input sample. Lossy: accept a controlled reconstruction difference to save more bits. For robotics inspection, never assume that a file visually pleasing to humans preserves the exact pixel evidence needed for tiny-defect detection.

11 Industrial Machine Vision

11.1 Inspection, sorting and robot guidance

An industrial vision system may locate parts, read markings, estimate dimensions, inspect defects and guide a pick-and-place robot. A conveyor sorting station typically needs stable lighting and a trigger so that the image corresponds to a known conveyor position. The system should reject uncertain or blurry detections for reinspection rather than committing to an unverified robot motion.

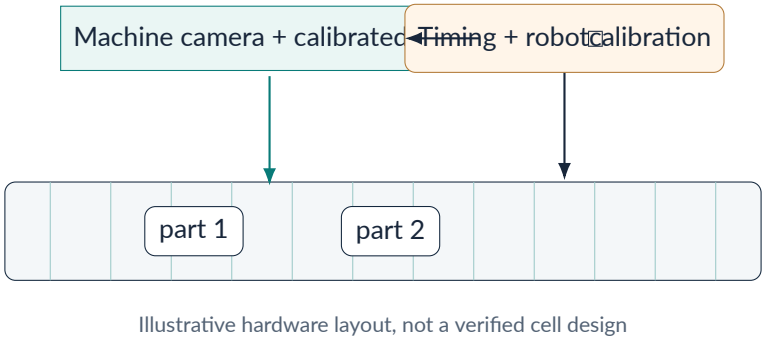

Figure 11. Original camera-to-conveyor concept showing the trigger, controlled illumination, moving parts and calibrated downstream actuation.

11.2 Converting a centroid to a robot waypoint

For a flat calibrated plane, a homography H can map homogeneous image coordinates $\mathbf{p} = (u, v, 1)^T$ to a planar world coordinate up to scale. With a locally rectified simple pixel scale, $X = s_x(u - u_0)$ and $Y = s_y(v - v_0)$, but this shortcut is not valid for arbitrary 3D geometry or uncalibrated perspective. The robot frame also requires a verified extrinsic transformation.

Rectified 2D mapping

With image origin at pixel $(u_0, v_0) = (320, 240)$, scale $s_x = s_y = 0.2$ mm/pixel and detected centroid $(u, v) = (350, 220)$, the corresponding plane offsets are $X = (350 - 320) \cdot 0.2 = 6$ mm and $Y = (220 - 240) \cdot 0.2 = -4$ mm. Signs depend on the chosen camera/world axis convention. A robot command needs the camera-to-robot calibration as well as a collision check.

11.3 End-to-end timing and uncertainty

Image localization error includes pixel measurement uncertainty, lens distortion, calibration error, part-height variation and timing delay. A moving belt at 0.5 m/s travels $0.5 \cdot 0.12 = 0.06$ m, or 60 mm, in 120 ms. The controller must use a properly time-stamped moving-part model rather than a stale centroid.

Industrial checklist
Use repeatable illumination; verify exposure and focus; calibrate at operating geometry; characterize false rejects and missed defects; sample compressed and uncompressed images; test under shifts/loads; include hardware interlocks; and document what happens when vision confidence is low.

12 Solved Practice and Answer Bank

1. Define binary, 8-bit grayscale and 8-bit/channel RGB images. For 640×480 RGB, find raw bytes. **Answer:** 921,600 B.
2. State the image coordinate system used here. **Answer:** x column rightward, y row downward, top-left origin zero.
3. Find the centroid of $(1,1),(3,1),(2,4)$. **Answer:** $(2,2)$.
4. A gray row has weights $1,2,3$ at $x=0,1,2$. Find $\bar{x}$. **Answer:** $8/6 = 4/3$.
5. Why does RGB need a centroid policy? **Answer:** Three channel values do not uniquely define a spatial scalar weight.
6. Use a mean 3×3 filter on one isolated 90-valued centre. **Answer:** 10 at its centre output.
7. Threshold $[35,120,180]$ at $T=100$ with $>$ as foreground. **Answer:** $[0,1,1]$.
8. With $G_x=3, G_y=4$, compute gradient magnitude. **Answer:** 5 (Euclidean).
9. Name the four Canny stages. **Answer:** smoothing, gradient, nonmaximum suppression, double threshold plus hysteresis (the last two are often considered a combined stage).
10. Distinguish segmentation by threshold, region, edge and K-Means. **Answer:** intensity cut, seeded connectivity, contours, feature-space clusters.

11. Compute SSD for $T = (2, 4, 6), I = (1, 5, 7)$. **Answer:** 3, smaller is better.
12. Evaluate ZNCC for $T = (1, 2, 3)$ and $I = (2, 4, 6)$. **Answer:** $+1$; invalid if variance is zero.
13. RLE encode $[0, 0, 0, 1, 1, 0]$. **Answer:** $(0, 3), (1, 2), (0, 1)$.
14. State why DCT is not the sole cause of JPEG loss. **Answer:** finite quantization removes coefficient information.
15. Explain baseline JPEG versus JPEG 2000. **Answer:** baseline JPEG commonly uses block DCT; JPEG 2000 uses wavelet-based coding and can offer lossless options.
16. A moving belt travels at 0.5 m/s during 120 ms delay. **Answer:** 60 mm, so timestamp compensation is critical.

Frequent exam errors
Treating a pixel centroid as a complete grasp pose; using x and y reversed; assuming RGB has one natural centroid; using Canny as a single threshold filter; saying SSD should be maximized; forgetting ZNCC zero variance; assuming all JPEG is lossless; and treating an uncalibrated pixel displacement as millimetres.

OpenCV: a first experiment

The pseudocode below is a test of *image analysis*, not a robot-motion controller. Use your own known test image; an empty mask is a valid failed detection.

```
import cv2
img = cv2.imread("test_part.png", cv2.IMREAD_GRAYSCALE)
if img is None:
    raise FileNotFoundError("test_part.png")
_, mask = cv2.threshold(img, 100, 255,
                        cv2.THRESH_BINARY)
m = cv2.moments(mask, binaryImage=True)
if m["m00"] > 0:
    centre = (m["m10"] / m["m00"],
              m["m01"] / m["m00"])
else:
    centre = None  # No detected object
edges = cv2.Canny(img, 50, 100)
# Never send an uncalibrated 'centre' to a robot.
```

Three things to verify before a demo
(1) Show the binary mask as well as the original image; the selected object might be background. **(2)** Change lighting and check if the centroid remains stable. **(3)** Record x as column and y as row, then calibrate the image frame to robot coordinates before movement.

Sources and Figure Provenance

Official supplied syllabus: SPPU TE Robotics & Automation, 2024 Pattern, PCC301-ROA, Unit V, user-provided *TE Syllabus.pdf*, physical PDF p. 15. All named Unit V topics are explicitly mapped above.

Technical sources independently consulted:

1. OpenCV, *Image Moments*: https://docs.opencv.org/4.x/d0/d49/tutorial_moments.html
2. OpenCV, *Sobel Derivatives*: https://docs.opencv.org/4.x/d2/d2c/tutorial_sobel_derivatives.html
3. OpenCV, *Canny Edge Detector*: https://docs.opencv.org/4.x/da/d5c/tutorial_canny_detector.html
4. OpenCV, *Template Matching*: https://docs.opencv.org/4.x/de/da9/tutorial_template_matching.html
5. OpenCV, *Thresholding*: https://docs.opencv.org/4.x/d7/d4d/tutorial_py_thresholding.html
6. JPEG Committee, *JPEG 2000*: https://jpeg.org/jpeg2000/

7. ITU-T, JPEG baseline standard T.81: https://www.itu.int/itu-t/recommendations/rec.aspx?id=2633&lang=en

All embedded figures are original TikZ vectors and original programmatically generated raster figures, produced from disclosed synthetic arrays. No third-party image files or web photographs are embedded. A Wikimedia Commons CC0 conveyor photo was located for possible future use, but the image could not be downloaded because network transfer to the build environment failed; the image is therefore not part of this PDF.

Verification scope: Formula checks and PDF geometry checks evaluate selected calculations and legibility signals, not final industrial calibration, licence audits of absent assets, camera/robot hardware tests or independent academic proofreading.

UNIT V | INDEPENDENT NUMERICAL QA

Zenith Study Hub | Artificial Intelligence for Robotics | PCC301-ROA

Machine Vision in Robotics and Intelligent Systems | SPPU TE Robotics and Automation | 2024 Pattern

This supplement preserves the 21-page original teaching guide and adds complete question-bank computations. Source: supplied AIFR Unit-Wise Question Bank, Unit V Q2, Q4, Q6, Q8, Q10, Q12. Review candidate only; no faculty certification or production deployment.

1 | Coverage and notation audit

Syllabus requirement	Location in original teaching guide
Binary / grayscale / RGB images, image resolution and centroids	Pages 3-6: definitions, memory, binary/weighted grayscale/RGB centroids
Smoothing, sharpening, threshold, gradients and edge detectors	Pages 7-11: mean/Gaussian, sharpening, Sobel, Prewitt, Roberts, LoG, Canny
Threshold/region/edge/K-Means segmentation	Pages 11-12: method comparison and short clustering example
SSD, NCC; lossless and lossy image compression	Pages 13-16: SSD, ZNCC, RLE, Huffman, LZW, JPEG/DCT/Wavelets
Machine vision sorting, inspection and guidance	Pages 17-18: cell concept, calibration, timing and robot controls

Coordinate convention: The original guide defines zero-based x=column, y=row. Question-bank Q2 does not explicitly state an origin. We provide a 1-based exam-table result and a zero-based equivalent. For matching questions, translation (2,3) is expressly 1-based Row 2, Column 3, so the top-left patch index is (row 1, col 2) in zero-based Python.

Q2 | Weighted grayscale centroid (3 x 3)

Given grayscale intensities by rows: [10,20,30], [40,50,60], [70,80,90]. Use column x and row y each numbered 1, 2, 3. With moments M00=sum I, M10=sum xI, M01=sum yI:

Moment	Calculation	Value
M00	10+20+30+40+50+60+70+80+90	450
M10	1(10+40+70)+2(20+50+80)+3(30+60+90)	960
M01	1(10+20+30)+2(40+50+60)+3(70+80+90)	1080

Answer in 1-based coordinates: Cx=960/450=**2.133333**; Cy=1080/450=**2.400000**. If an examiner expects zero-based image coordinates, subtract one from each axis: **(1.133333,1.400000)**. Neither answer is interchangeable without declaring the origin.

Q4 | Prewitt gradient on the 3 x 3 image

Use the same image matrix as Q2. Choose the image-correlation sign convention with horizontal Prewitt kernel Px=[[-1,0,1],[-1,0,1],[-1,0,1]] and vertical kernel Py=[[-1,-1,-1],[0,0,0],[1,1,1]].

Gx=(-10+30)+(-40+60)+(-70+90)=**60**. Gy=-(10+20+30)+(70+80+90)=**180**. Euclidean magnitude G=sqrt(Gx^2+Gy^2)=sqrt(3600+32400)=**sqrt(36000)=189.736660**.

Sign convention: a true mathematical convolution flips the masks and reverses gradient signs; magnitude is unchanged. Some exams use the L1 approximation |Gx|+|Gy|=240, but it is not the requested Euclidean magnitude.

Q6 | Zero-mean normalized cross-correlation (ZNCC)

Template T: [4 2 5] / [3 1 2] / [7 6 3]. Image rows: [1 2 4 6 5 3] / [3 5 7 8 4 2] / [6 8 5 9 7 4] / [5 3 6 2 1 3] / [4 6 8 7 9 5].

At translation (2,3), 1-based top-left Row 2 Col 3, extract patch P: [7 8 4] / [5 9 7] / [6 2 1]. This is NOT the row-3/col-2 patch.

Mean(T)=33/9=11/3=3.666667; Mean(P)=49/9=5.444444.

Using sum((T-mean T)(P-mean P)) / sqrt(sum((T-mean T)^2) sum((P-mean P)^2)):

Numerical component	Verified value
Centred cross-product sum	-62/3 = -20.666667
Template sum of squared deviations	32
Patch sum of squared deviations	524/9 = 58.222222
Denominator	sqrt(32 x 524/9) = 43.163771
ZNCC Q6	**-0.478796599**

Interpretation: negative ZNCC implies an anticorrelated brightness pattern relative to this template at the specified location. This score is not a probability or a quality guarantee. The original guide calls this **zero-mean** normalized cross-correlation; ordinary uncentred normalized correlation is a different formula.

Q10 | Second NCC numerical with distinct data

Template T: [3 1 4] / [2 5 6] / [7 3 2]. Image rows: [1 2 3 4 5 6] / [2 4 6 8 1 3] / [5 7 9 2 4 6] / [3 5 7 1 2 4] / [6 8 1 3 5 7].

Extract the patch from 1-based Row 2, Column 3: [6 8 1] / [9 2 4] / [7 1 2]. Means: mean(T)=33/9=3.666667 and mean(P)=40/9=4.444444.

Numerical component	Verified value
Centred cross-product sum	-26/3 = -8.666667
Template sum of squared deviations	32
Patch sum of squared deviations	704/9 = 78.222222
Denominator	sqrt(32 x 704/9) = 50.031101
ZNCC Q10	**-0.173225582**

Q10 is a different image and a different template from Q6. Its higher NCC is still negative: less anticorrelated, but not a strong positive match.

Q8 | Run-length encoding of first 5 x 6 binary image

RLE policy: process each image row separately, left to right. Store explicit (pixel value, run length) pairs. Reset the run at each new row, do not join rows even if the end/start symbols match.

Row	Original row	Encoded (value,length)
1	1 1 1 0 0 0	(1,3) (0,3)

Row	Original row	Encoded (value,length)
2	1 1 0 0 0 1	(1,2) (0,3) (1,1)
3	1 0 0 1 1 1	(1,1) (0,2) (1,3)
4	1 0 1 1 0 0	(1,1) (0,1) (1,2) (0,2)
5	0 0 0 1 1 1	(0,3) (1,3)

Count: 2+3+3+4+2 = **14 runs**, yielding **28 stored numerical fields** in the explicit pair format. Source data contain 30 binary pixel symbols. This field-count comparison is not a byte-level compression ratio.

Q12 | Run-length encoding of second 5 x 6 image

Row	Original row	Encoded (value,length)
1	1 1 0 0 1 1	(1,2) (0,2) (1,2)
2	1 0 0 1 1 0	(1,1) (0,2) (1,2) (0,1)
3	0 0 1 1 1 0	(0,2) (1,3) (0,1)
4	1 1 1 0 0 0	(1,3) (0,3)
5	0 1 1 1 0 1	(0,1) (1,3) (0,1) (1,1)

Count: 3+4+3+2+4 = **16 runs**, yielding **32 stored numerical fields**. Thus this explicit RLE representation uses more fields than the 30 input symbols. Proper compression depends on packed bitstream syntax, run-length coding and headers.

Compression caveat | What an examiner should state

The question asks for the number of values stored. Under explicitly stored (value,length) pairs it is twice the number of runs. If a format stores the start symbol followed by alternating run lengths, the answer differs. Without a defined representation, there is **no unique byte-size figure**.

Independent source-example arithmetic checks

Guide section	Verified source example
Raw RGB image, page 4	640 x 480 x 3 = 921,600 bytes; 0.9216 MB and about 0.8789 MiB
Binary/gray/RGB centroid, pages 5-6	Binary=(2,2); weighted gray x=4/3; illustrative channel-sum RGB x=1.5
Mean / sharpening filters, page 8	90/9=10; 5 x 130 - 4 x 100 = 250
Sobel gradient, page 10	Gx=80 and Gy=0 under specified correlation convention
K-Means segmentation, page 12	[10,20,200,220] yields centroid means (15,210)
SSD and ZNCC, page 13	SSD 3; ZNCC of (1,2,3) vs (2,4,6) is +1
Huffman / RLE, pages 14-15	Example RLE has 3 runs; Huffman length 14/8=1.75 bits/symbol before overhead
JPEG quantization, page 16	round(23/10)=2, reconstructed coefficient 20
Pixel mapping / latency, pages 17-18	(350,220) maps to (+6,-4) mm under stated scaling; 0.5 m/s x 0.12s=60mm

Editorial QA observations from rendered original

The 21-page PDF has consistent A4 page size, visually rich original synthetic diagrams and stable running headers. A paragraph about NCC at the bottom of page 13 spills into page 14 as a dangling fragment ("trary illumination effects..."). This is a **reading-flow defect**, not an arithmetic error. The new candidate appends worked solutions but does not pretend that reflowing the original teaching text is complete.

The page-7 industrial vision flow and page-17 industrial machine vision drawing are conceptual diagrams, not verified factory engineering diagrams. Centroids, interpolation and camera-to-robot calibration should always distinguish image plane from world frame.

Release conditions and provenance

Official topic authority: supplied TE Syllabus.pdf, PCC301-ROA, Unit V on printed page 15. Numerical question authority: supplied AIFR_Unit_Wise_Question_Bank.pdf, Unit V Q2/Q4/Q6/Q8/Q10/Q12. The original PDF provides the theoretical teaching guide and illustrative examples; this supplement adds question-bank-specific numericals.

Pending: correct the original page-13/page-14 paragraph flow, inspect every original formula and diagram at high resolution, complete accessibility review, and obtain independent academic approval. This merged review edition is not certified for publication and is not deployed to the Zenith website.