RC! 26

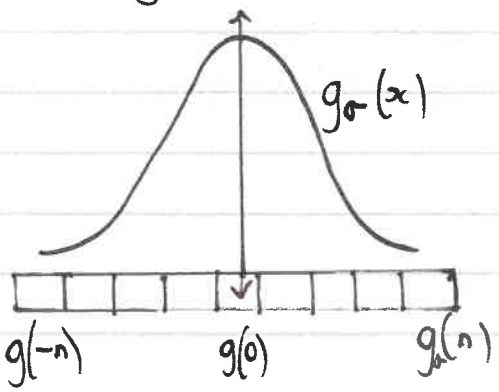
4F12 Computer Vision (2026 solutions)

Q1(a)(i)

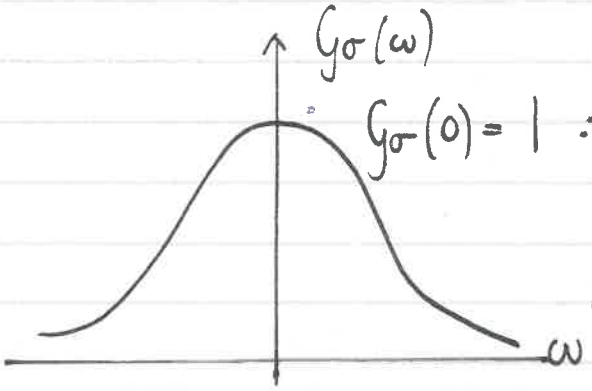
$$S_\sigma(x,y) = \sum_{-n}^{+n}\sum_{-n}^{+n} I(x-u, y-v)\, g_\sigma(u)\, g_\sigma(v) \qquad (2)$$

where $g_\sigma(x) = \frac{1}{\sqrt{2\pi}\,\sigma} e^{-\frac{x^2}{2\sigma^2}}$

and the gaussian kernel is sampled at $2n+1$ locations (1)



(ii) Fourier transform of $g_\sigma(x) \xrightarrow{F.T.} G_\sigma(\omega) = e^{-\frac{\omega^2\sigma^2}{2}}$



$G_\sigma(0) = 1 \therefore \int_{-\infty}^{\infty} g_\sigma(x)\,dx = 1$ (1)

$G_\sigma(\infty) \to 0$ (1)

Q1a (iii)

We need to show/find:

$$g_{\sigma_3}(x) = g_{\sigma_1}(x) * g_{\sigma_2}(x)$$

Taking fourier transforms:

$$G_{\sigma_3}(\omega) = G_{\sigma_2}(\omega) \times G_{\sigma_1}(\omega)$$

$$= e^{-\frac{\omega^2\sigma_1^2}{2}} \cdot e^{-\frac{\omega^2\sigma_2^2}{2}}$$

$$= e^{-\frac{\omega^2(\sigma_1^2+\sigma_2^2)}{2}}$$

$$\therefore g_{\sigma_3}(x) = \frac{1}{\sqrt{\sigma_1^2+\sigma_2^2}\sqrt{2\pi}} \cdot e^{-\frac{x^2}{2(\sigma_1^2+\sigma_2^2)}}$$

$$\underline{\sigma_3^2 = \sigma_1^2 + \sigma_2^2} \qquad (2)$$

In image pyramid for efficient low-pass filtering we apply a small incremental blur

$$g(\sigma_{i+1}) = g(\sigma_i) * g(\sigma_{Ki}) \qquad \text{where } \sigma_{i+1} = 2^{1/s}\sigma_0$$

$$\therefore \sigma_{i+1}^2 = \sigma_i^2 + \sigma_{Ki}^2 \Rightarrow 2^{\frac{2}{s}}\sigma_i^2 = \sigma_i^2 + \sigma_{Ki}^2$$

$$\underline{\sigma_{Ki}^2 = \left(2^{\frac{2}{s}} - 1\right)\sigma_i^2} \qquad (1)$$

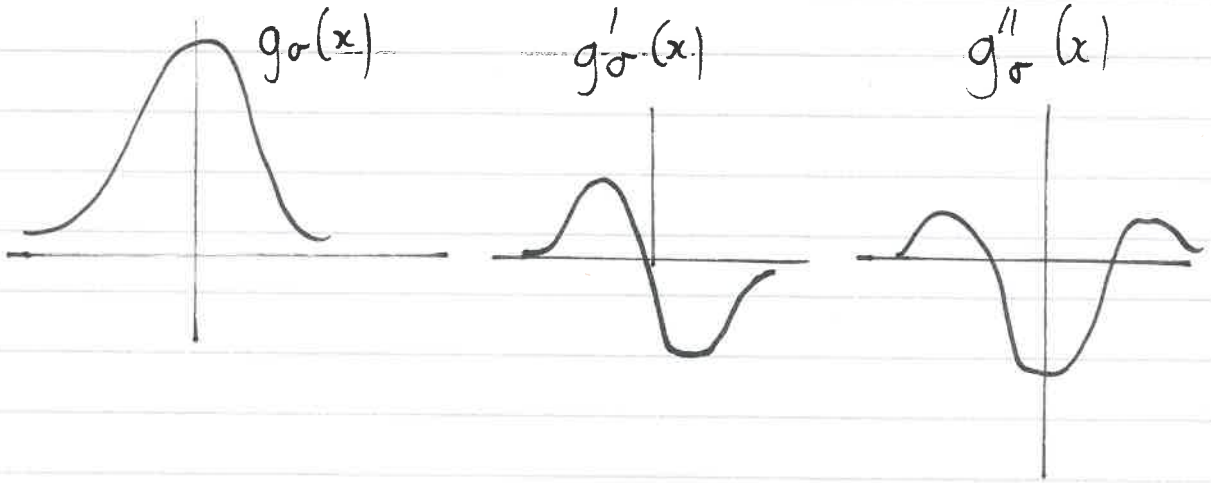
Q(b)(i).

$$g_\sigma(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-x^2}{2\sigma^2}}$$

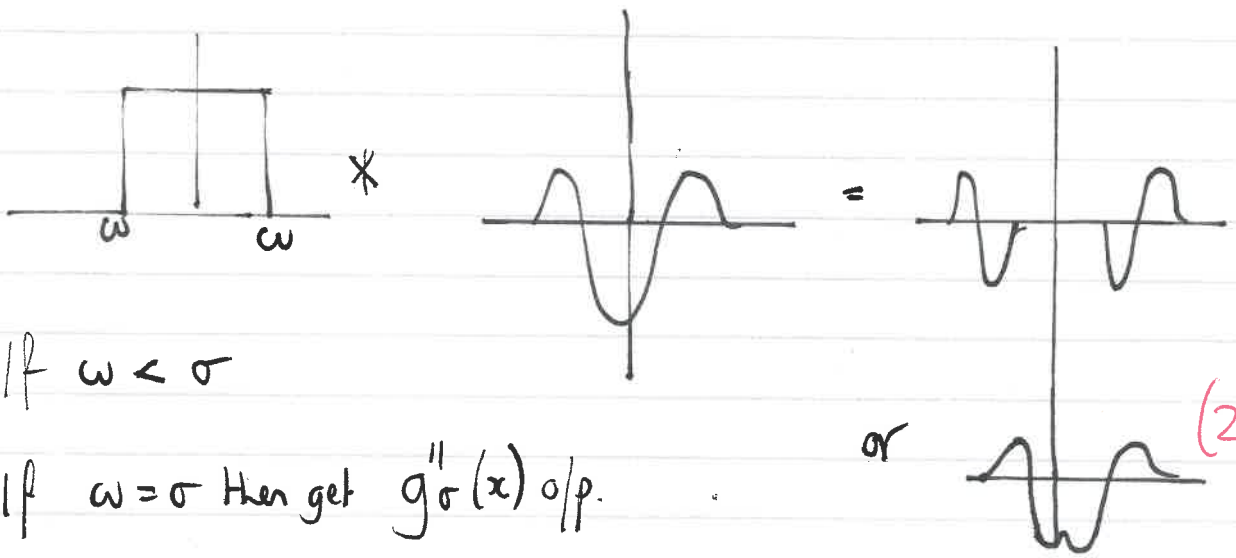
$$g'_\sigma(x) = \frac{1}{\sigma\sqrt{2\pi}}\left[-\frac{2x}{2\sigma^2} e^{\frac{-x^2}{2\sigma^2}}\right] = \frac{-x}{\sigma^2} g_\sigma(x)$$

$$g''_\sigma(x) = -\frac{1}{\sigma^2} g_\sigma(x) + \frac{x^2}{\sigma^4} g_\sigma(x)$$

$$= \left[\frac{x^2-\sigma^2}{\sigma^4}\right] g_\sigma(x) \qquad (2)$$



(ii)



If $\omega < \sigma$

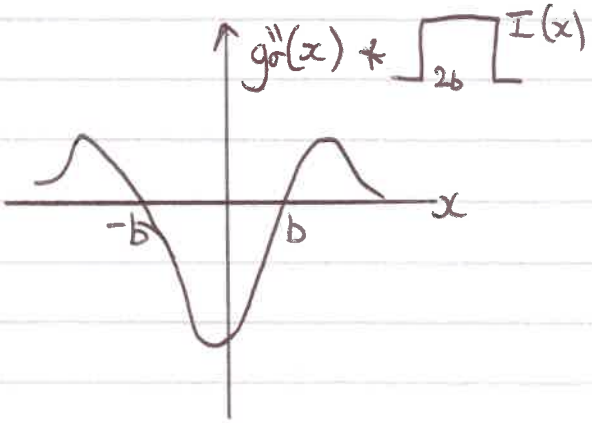
If $\omega = \sigma$ Then get $g''_\sigma(x)$ o/p. (2)

Q1 b(iii)

Maximum response when $b = \sigma$ (1D bar 

) (1)

or $r = \sqrt{2}\sigma$ (2D circular blob) (1)



b (iv) The scale-normalised Laplacian of a Gaussian

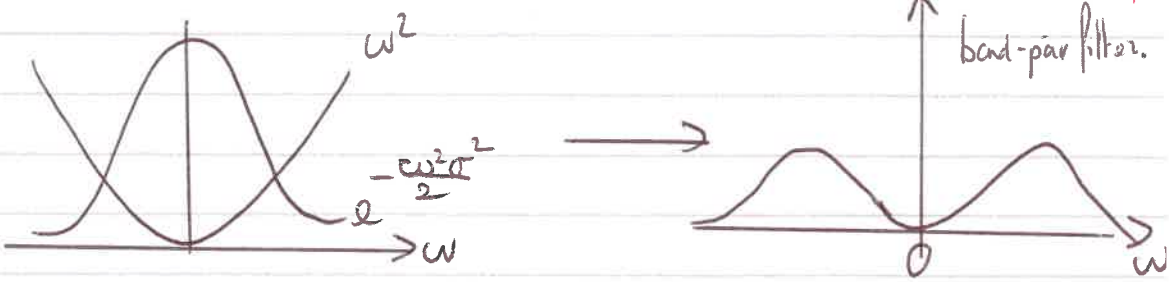
$$\sigma^2 \nabla^2 g_\sigma(x) \approx g_{k\sigma}(x) - g_\sigma(x) \text{ i.e. DOG}$$

To show it is a band-pass filter, look at Fourier Transform

$$g_\sigma(x) \xleftrightarrow{FT} e^{-\frac{\omega^2\sigma^2}{2}}$$

$$g''_\sigma(x) \longleftrightarrow -\omega^2 e^{-\frac{\omega^2\sigma^2}{2}}$$

(2)



Q1(c) (i) Edge — Discontinuity in $I(x,y)$

Compute $S_\sigma(x,y)$ and max local max in $\nabla S_\sigma(x,y)$

Localize as zero-crossing of $\nabla^2 S_\sigma(x,y)$ (2)

(ii) Corners

Auto-correlation is well-defined and a local peak

Compute $S_\sigma(x,y)$, $\frac{\partial S}{\partial x}$ and $\frac{\partial S}{\partial y}$

Compute 3 images and smooth/blur by σ_I

$$S_x^2 = \left|\frac{\partial S}{\partial x}\right|^2 \quad S_x S_y \left|\frac{\partial^2 S}{\partial x \partial y}\right| \quad S_y^2 = \left|\frac{\partial^2 S}{\partial y}\right|^2$$

For each pixel compute A

$$A = \begin{bmatrix} \langle S_x^2 \rangle & \langle S_y^2 \rangle \\ \langle S_x S_y \rangle & \langle S_x S_y \rangle \end{bmatrix}$$

where $\langle\ \rangle$ is σ_I window/smoothed values

Corners defined at $\det A - k(\text{trace } A)^2 >$ threshold

(2)

Q2

(a) Pin-hole camera with no non-linear distortion (ie. central projection to a plane) (1)

$(u,v) \longleftrightarrow (x_1, x_2, x_3)$

$(X,Y,Z) \longleftrightarrow (X_1, X_2, X_3, X_4)$ (1)

cartesian — homogeneous

(b)(i) Perspective projection of (X,Y,Z) to (u,v)

$$u = \frac{x_1}{x_3} = \frac{p_{11}X + p_{12}Y + p_{13}Z + p_{14}}{p_{31}X + p_{32}Y + p_{33}Z + p_{34}}$$

$$v = \frac{x_2}{x_3} = \frac{p_{21}X + p_{22}Y + p_{23}Z + p_{24}}{p_{31}X + p_{32}Y + p_{33}Z + p_{34}}$$ (1)

and $S = x_3$ = distance to optical axis $= p_{31}X + p_{32}Y + p_{33}Z + p_{34}$ (1)

(b)(ii) $\underline{VP}_X = \begin{bmatrix} p_{ij} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} p_{11} \\ p_{21} \\ p_{31} \end{bmatrix}$

$\underline{VP}_Y = \begin{bmatrix} p_{ij} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} p_{12} \\ p_{22} \\ p_{32} \end{bmatrix}$

$\underline{L}_{horizon} = \underline{VP}_X \times \underline{VP}_Y = \begin{bmatrix} p_{11} \\ p_{21} \\ p_{31} \end{bmatrix} \times \begin{bmatrix} p_{12} \\ p_{22} \\ p_{32} \end{bmatrix} = \begin{vmatrix} i & j & k \\ p_{11} & p_{21} & p_{31} \\ p_{12} & p_{22} & p_{32} \end{vmatrix}$ (2)

1(c)

(i) 12 homogeneous parameters p_{ij}
Scale is arbitrary, hence 11 unknowns.

Real camera has R, T, $K = \begin{bmatrix} k_u f & 0 & u_0 \\ 0 & k_v f & v_0 \\ 0 & 0 & 1 \end{bmatrix}$
3 dof 3 dof 4 dof

$\therefore$ 10 parameters (d.f freedom since $k_u f$, $k_v f$) (1)

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = [K][R \mid T] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Each (u_i, v_i) & known $(X_i Y_i Z_i)$ gives 2 independent equations

$\therefore$ Need $N \geq 5.5$ or 6 points (1)

(ii) Calibrate camera to recover known 3D object's position (T) and orientation R.

- N known points measure (u_i, v_i) to give 2 equations
- Set up $\underset{2N \times 12}{[\ A\ ]} \underset{12 \times 1}{\begin{bmatrix} p_{11} \\ \vdots \\ p_{34} \end{bmatrix}} = \underline{0}$ (1)
- Solve by linear least squares $\lambda_1 \leq \frac{\underline{p}^T A^T A \underline{p}}{\underline{p}^T \underline{p}} \leq \lambda_{12}$ (1)
  Find $\underline{p}$ which minimises square errors. $\underline{p}$ is eigenvector corresponding to λ_1

1c(iii) cont.

— Solve by non-linear optimisation

$$\min_{P} \sum (u_i - \hat{u}_i)^2 + (v_i - \hat{v}_i)^2 \quad (1)$$

where (u_i, v_i) are measurements and $\begin{bmatrix} s\hat{u}_i \\ s\hat{v}_i \\ s \end{bmatrix} = \begin{bmatrix} P_{ij} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$ (1)

— Decompose $\underline{P} = [KR \; KT]$

to find $\underline{KR}$ by $\underline{RQ}$ decomposition $\Rightarrow$ $\underline{K}$ and $\underline{R}$

and $T = K^{-1} \begin{bmatrix} P_{14} \\ P_{24} \\ P_{34} \end{bmatrix}$ (1)

Q1c(d) Under weak perspective $Z_c = Z_{AV}$ or $S_{NP} = Z_{AV}$

(i) $\therefore \begin{bmatrix} su \\ sv \\ s \end{bmatrix} = 3 \begin{bmatrix} P_{ij} \; (4) \\ 0 \; 0 \; 0 \; Z_{av} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$

instead of $S = p_{31}X + p_{32}Y + p_{33}Z + p_{34}$

$$\begin{bmatrix} u \\ v \end{bmatrix} = 2 \begin{bmatrix} P_{ij} \; (4) \\ 2 \times 4 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Q2 d(ii)

$\frac{\Delta z}{Z} \ll 1$ and small field of view

error due to weak perspective

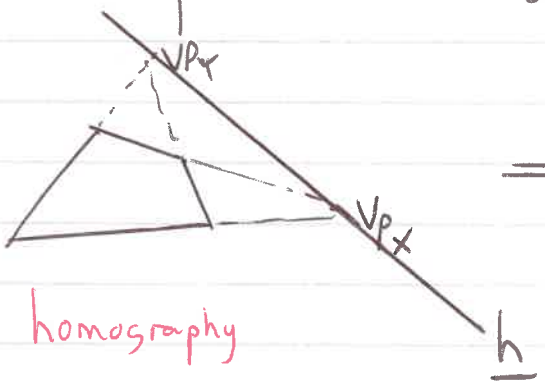
$$(u - u_A) = \frac{\Delta z}{z}(u - u_0) \qquad (2)$$

$$(v - v_A) = \frac{\Delta z}{z}(v - v_0)$$

Advantages

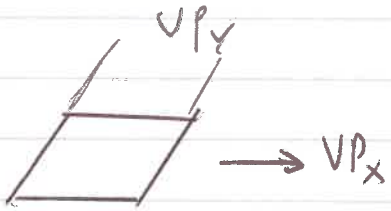
- projection becomes linear $S = Z_{AV} = \text{constant}$
- least-squares is now optimal (2)
- Only need 4 points to calibrate

d(iv) X-Y plane under homography $\Longrightarrow$ affine transformation



homography

$\Longrightarrow$



VP_s and horizon is sent to ∞

affine

(2)

Crib for 4F12 Computer Vision 2026

Version: MJ/3

Marking guidelines: words in **blue** are key words that should be mentioned in long-form answers. Notes in **magenta** are guidelines to use when marking the answer.

Question 3

a)

i)

$$\begin{aligned} I_1^{(1,1,1)} =& W_0^{(1,0,0)} I_0^{(0,0)} + W_0^{(1,0,1)} I_0^{(0,1)} + W_0^{(1,0,2)} I_0^{(0,2)} + \\ & W_0^{(1,1,0)} I_0^{(1,0)} + W_0^{(1,1,1)} I_0^{(1,1)} + W_0^{(1,1,2)} I_0^{(1,2)} + \\ & W_0^{(1,2,0)} I_0^{(2,0)} + W_0^{(1,2,1)} I_0^{(2,1)} + W_0^{(1,2,2)} I_0^{(2,2)} \end{aligned}$$

ii)

Need to mention at least 2 types of padding and how they work, and the fact that it is a padding of 1

This is achieved by using **padding** of 1 on all sides, so that when the filter is applied at the **edges and corners** of the image the padding provides values for the **missing pixels**. There are three kinds of padding that are normally used: **constant value**, **reflect**, and **repeat**. Constant value uses a provided value for all missing pixels. Reflect reflects values over the edge. Repeat repeats the edge value.

iii)

This filter will have a strong positive response on edges oriented at 45° degrees, and a strong negative response on edges oriented at 135°.

b)

i)

The response needs to highight the advantages of incorporating human expertise, and the disadvantage of limiting the space of what the network can learn. Any valid scenario they describe is acceptible.

The advantage of using engineered features as input to a CNN is that it allows **human domain expertise** to be provided for use by the CNN. For some kinds of data, this may be necessary, as it does not suit itself well to processing by the network, for example if the data contains **identifiable information**. The downside is that if that if only engineered features are provided and not

the original data then the network cannot learn new features appropriate to the problem but is limited by those that were engineered by the human operators. Aside from the scenario already mentioned, some possible scenarios include: data scarcity, data that is not recorded as floating point numbers, access to hardware acceleration.

ii)

One possibility is that W_1 is $M \times 3 \times 3 \times N$ and applied with no padding. This is just one possibility though, for example it could be 5×5 and applied with padding of 1.

iii)

The key concept to communicate here is that the convolutional layer assembles low-level image features into a more semantically coherent entity, and the pooling layer reduces the image dimensionality as the semantic structure increases. The exact term used, e.g., parts, does not matter provided they make their meaning clear.

The filter bank helps extract **image features**. The next step is to assemble those features into **parts**. The convolutional layer learns how to combine the **low-level features** by convolving over them with **three-dimensional kernels**, and the responses are then **reduced in dimensionality** by the pooling layer. This relaxes the positional requirement on the parts, and also allows the **dimensionality of the image to be reduced** as semantic depth and complexity increases.

iv)

Needs to mention that fine-tuning adapts weights to a specific problem, and that there is the danger of catastrophic forgetting. Reducing the learning rate on the fine-tuning layer or a similar mitigation for handling this problem needs to be mentioned.

Fine-tuning is when **pre-trained weights** are adapted to a new problem by small changes in their values via **back-propagation**. The engineered filters in W_0 may not be perfectly suited to the problem at hand, and this allow them to get minor adjustments to make them better. The downside is that the network may end up **forgetting them** entirely and having to relearn them from scratch, but this can be avoided by **lowering the learning rate**.

c)

This module is performing **positional embedding**. Each dimension n in I_1 corresponds to a **different frequency**, and the **frequency response of the pixel location** is added to the pixel value. A Vision Transformer uses positional embeddings like this to **encode the position of a pixel or patch** in the original image, as it treats all patches as a **stream of tokens** without any 2D

structure. A CNN performs **structural convolution**, in which early layers preserve the **2D structure** of the image, thus **implicitly encoding** position.

d)

They need to explain that the patches are produced from the training dataset without supervision, and that learned embeddings are neural nets which convert patches to a latent space. Some version of patch dictionaries being good for decoding and learned embeddings being good for encoding is the key thing to communicate when comparing and constrasting.

A patch dictionary consists of a **codebook of patches** produced from **unsupervised** analysis of the **training dataset**, for example by **k-means clustering**. Each patch in the grid is then assigned the **integer value** of the codeword **nearest to it** in the dictionary, which is then turned into a **one-hot vector** that is used as the **token**. The output tokens are generated from **probability distributions over the codebook**. Because the output is a distribution over codewords, this makes **decoding** using techniques like **beam search** result in better output, thus making patch dictionaries a better fit for Transformers which include a **Decoder block**.

By contrast, a learned embedding is a **neural net (MLP or CNN)** which takes the input patch and produces a vector in a **latent space**, which then acts as the token. This provides a richer representation than a patch dictionary, as it is learned based on the task itself and each patch can have its **own unique representation**. However, output tokens are **harder to interpret** unless the embedding is **invertible**. As such, learned embeddings are better suited to **Encoder-Only Transformers**.

Q4(a) Geometric matching constraint is the epipolar constraint

i.e In stereo vision, baseline and 2 rays are coplanar. ($\underline{p}' \cdot \underline{t}_x R \underline{p} = 0$)

(1)

Algebraic expression

$$\begin{bmatrix} u' & v' & 1 \end{bmatrix} \begin{bmatrix} & F & \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = 0$$

where

$$\underset{3\times3}{F} = \underset{3\times3}{K'^{-T}}\ \underset{3\times3}{T_x}\ \underset{3\times3}{R}\ \underset{3\times3}{K^{-1}}$$

and $T_x = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}$ skew-symmetric

$\underline{R} = \begin{bmatrix} 3\times3 \end{bmatrix}$ rotation matrix

(2)

4(b)(i) Algorithm to find feature points and descriptors:

(1)

- SIFT — find blob centres and scales by DOG from image pyramid
  - sample 16×16 at correct scale σ_i and dominant orientation
  - 128D descriptor from 16 HOGs for each 4×4 cell
    $\underline{x}_i$
  - invariant to scale, orientation and lighting and normalised to $|\underline{x}_i| = 1$

Algorithm for finding correspondences:

- features match if $\underline{x}_i$ and $\underline{x}_j$ agree
- Find euclean distance $\|\underline{x}_i - \underline{x}_j\|^2$
  $\underline{d}_1$

(2) $\dfrac{|\underline{d}_1 - \underline{x}_i|^2}{|\underline{d}_2 - \underline{x}_i|^2} < 0.7$ — Accept as match if closest match is much closer than second $\underline{d}_2$ closest match by RATIO test

Q4(c)(ii)

Recover 2 projection matrices from K, K', R and $\underline{t}$

$$\underline{P} = K[I|0] \quad \text{and} \quad \underline{P'} = K'[R|\underline{t}]$$

Each pair of correspondences gives 4 equation in 3 unknowns

$$u(p_{31}X + p_{32}Y + p_{33}Z + p_{34}) - (p_{11}X + p_{12}Y + p_{13}Z + p_{14}) = 0$$
$$v(p_{31}X + p_{32}Y + p_{33}Z + p_{34}) - (p_{21}X + p_{22}Y + p_{23}Z + p_{23}) = 0$$

and also for $(u'\ v')$ $\begin{bmatrix} su' \\ sv' \\ s \end{bmatrix} = \begin{bmatrix} \\ p'_{ij} \\ \\ \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$ (2 more equations of planes)

$$\therefore \underset{4}{\begin{bmatrix} & & \\ & A & \\ & & \end{bmatrix}} \underset{4\times 1}{\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}} = \underline{0}$$

Solve by least squares $\lambda_1 \leq \dfrac{\underline{X}A^TA\underline{X}}{X^TX} < \lambda_4$. (2)

This is algebraic soln but corresponds to triangulation of 2 rays (or intersection of 4 planes through the 3D point).

Q4(a) Geometric matching constraint is the epipolar constraint

i.e In stereo vision, baseline and 2 rays are coplanar. ($\underline{p}' \cdot \underline{t}_x R \underline{p} = 0$)

(1)

Algebraic expression

$$\begin{bmatrix} u' & v' & 1 \end{bmatrix} \begin{bmatrix} & F & \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = 0$$

where

$$\underset{3\times3}{F} = \underset{3\times3}{K'^{-T}}\ \underset{3\times3}{T_x}\ \underset{3\times3}{R}\ \underset{3\times3}{K^{-1}}$$

and $T_x = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}$ skew-symmetric

$\underline{R} = \begin{bmatrix} 3\times3 \end{bmatrix}$ rotation matrix

(2)

4(b)(i) Algorithm to find feature points and descriptors:

(1)

- SIFT — find blob centres and scales by DOG from image pyramid
  - sample 16×16 at correct scale σ_i and dominant orientation
  - 128D descriptor from 16 HOGs for each 4×4 cell
    $\underline{x}_i$
  - invariant to scale, orientation and lighting and normalised to $|\underline{x}_i| = 1$

Algorithm for finding correspondences:

- features match if $\underline{x}_i$ and $\underline{x}_j$ agree
- Find euclean distance $\|\underline{x}_i - \underline{x}_j\|^2$
  $\underline{d}_1$

(2) $\dfrac{|\underline{d}_1 - \underline{x}_i|^2}{|\underline{d}_2 - \underline{x}_i|^2} < 0.7$ — Accept as match if closest match is much closer than second $\underline{d}_2$ closest match by RATIO test

4(b)(ii). Each pair of correspondences (u,v) and (u',v') gives a single constraint on F

$$\begin{bmatrix} u' & v' & 1 \end{bmatrix} \begin{bmatrix} F \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = 0$$

$$\begin{bmatrix} u'u & u'v & u' & v'u & v'v & v' & u & v & 1 \end{bmatrix} = 0$$

∴ Need 8 points to estimate F by least-squares / SVD

$$\underset{8\times 9}{A}\ \underset{9\times 1}{f} = 0 \qquad (1)$$

In presence of outliers use RANSAC (2)

- match features over 2 views
- select 8 pairs randomly
- compute F using minimal subset
- check inliers
- repeat until maximum inliers

- perform least-squares on all N inliers

(iii) $\underset{N\times 9}{A}\ \underset{9\times 1}{f} = 0$ (1)

Solve by $\lambda_1 \leq \dfrac{f^T A^T A f}{f^T f} \leq \lambda_9$ (2)

Find f corresponding to smallest eigenvalue λ_1. (3)

Need to satisfy $\det F = 0$ Hence compute $F = [U]\Lambda[V]^T$ (4)
and reproject to $F = [U]\begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & 0 \end{bmatrix}[V]^T$

Q4 (c)

(i) $[E] = K'^T [F] K$ (1) from known K and K' and estimated $[F]$
3×3

SVD of E

$$E = U \Lambda V^T$$

where $R_1 = U \underset{W}{\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}} V^T$ { rotation matrix

or $R_2 = U \underset{W^T}{\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}} V^T$ { rotation matrix, reflected

(1)

and $\hat{T}_x = U \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} U^T$ skew-symmetric matrices

or $-\hat{T}_x = U \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} U^T$ (1)

4 solutions — Ambiguity and magnitude of $|T|$ unknown

Resolve ambiguity by setting up $P_1 = K[I|0]$
$P_2 = K'[R|t]$

and seeing which combination of $\pm t$ R_1 or R_2 gives +ve depth (ie features infront of camera). $|t|$ is still unknown. Need known baseline (1)

Q4(a) Geometric matching constraint is the epipolar constraint

i.e In stereo vision, baseline and 2 rays are coplanar. ($\underline{p}' \cdot \underline{t}_x R \underline{p} = 0$)

(1)

Algebraic expression

$$\begin{bmatrix} u' & v' & 1 \end{bmatrix} \begin{bmatrix} & F & \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = 0$$

where

$$\underset{3\times3}{F} = \underset{3\times3}{K'^{-T}}\ \underset{3\times3}{T_x}\ \underset{3\times3}{R}\ \underset{3\times3}{K^{-1}}$$

and $T_x = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}$ skew-symmetric

$\underline{R} = \begin{bmatrix} 3\times3 \end{bmatrix}$ rotation matrix

(2)

4(b)(i) Algorithm to find feature points and descriptors:

(1)

- SIFT — find blob centres and scales by DOG from image pyramid
  - sample 16×16 at correct scale σ_i and dominant orientation
  - 128D descriptor from 16 HOGs for each 4×4 cell
    $\underline{x}_i$
  - invariant to scale, orientation and lighting and normalised to $|\underline{x}_i| = 1$

Algorithm for finding correspondences:

- features match if $\underline{x}_i$ and $\underline{x}_j$ agree
- Find euclean distance $\|\underline{x}_i - \underline{x}_j\|^2$
  $\underline{d}_1$

(2) $\dfrac{|\underline{d}_1 - \underline{x}_i|^2}{|\underline{d}_2 - \underline{x}_i|^2} < 0.7$ — Accept as match if closest match is much closer than second $\underline{d}_2$ closest match by RATIO test

Q4(d)

Under pure rotation there is no baseline ($\underline{t}=0$) and it is impossible to triangulate to recover (X,Y,Z) — (1)

Very small motion will lead to a small baseline and inaccurate triangulation and noise-sensitive reconstruction. (1)