4F12 Computer Vision (2024) – Solutions

Q1(a)(i)

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

where $g_\sigma(x) = \dfrac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}$ and is sampled $(2n+1)$ pts

(2 marks)

(ii) $S(x,y,\sigma_i)$ is family of low-pass filtered images (scale space)

1 – sample logarithmically $S(x,y,\sigma_i)$ $\sigma_i = \sigma_0 2^{i/s}$
σ_0, $\sigma_1 = \sigma_0 2^{\frac{1}{3}}$, $\sigma_2 = \sigma_0 2^{2/3}$, $\sigma_3 = 2\sigma_0$.

– subsample after s images $(s=3)$ to $\frac{1}{4}$ size.
(octave)

1 – In each octave use incremental blurs
eg from σ_i to σ_{i+1}
$\sigma_{K_i} = \sigma_i\sqrt{2^{2/3}-1}$

(need σ_0, σ_{K_1}, σ_{K_2} and σ_{K_3}) 4 kernels

These are used in all octaves (no need to compute new kernels)

Q1 (a)(ii)(cont)

$\sigma_{K_0} = \sigma_0\sqrt{2^{\frac{2}{3}}-1}$

	size				
Octave 1	512×512	$\boxed{\sigma_0}$	$\boxed{\sigma_0 2^{\frac{1}{3}}}$	$\sigma_0 2^{\frac{2}{3}}$	$2\sigma_0$
octave 2	256×256	$2\sigma_0$	$2\sigma_0 2^{\frac{1}{3}}$	$2\sigma_0 2^{\frac{2}{3}}$	$4\sigma_0$
octave 3	128×128	$4\sigma_0$	$4\sigma_0 2^{\frac{1}{3}}$	$4\sigma_0 2^{\frac{2}{3}}$	$8\sigma_0$
octave 4	64×64	$8\sigma_0$	$8\sigma_0 2^{\frac{1}{3}}$	$8\sigma_0 2^{\frac{2}{3}}$	$16\sigma_0$
octave 5	32×32	$16\sigma_0$			

(4 marks)

(a)(iii)

Incremental filters
$$\left.\begin{aligned} \sigma_{K_0} &= \sigma_0\sqrt{2^{\frac{2}{3}}-1} \\ \sigma_{K_1} &= \sigma_0 2^{\frac{1}{3}}\sqrt{2^{\frac{2}{3}}-1} \\ \sigma_{K_2} &= \sigma_0 2^{\frac{2}{3}}\sqrt{2^{\frac{2}{3}}-1} \end{aligned}\right\}$$
re used in all octaves

Initial blur σ_0

$\therefore$ 4 distinct filters only

(2 marks)

Q1(a)(iv)

Kernel for $\frac{\partial S}{\partial x}$ $\quad \boxed{\tfrac{1}{2} \mid 0 \mid -\tfrac{1}{2}}$ 1D convolution

Kernel for $\frac{\partial^2 S}{\partial x^2}$ $\quad \boxed{+1 \mid -2 \mid +1}$ 1D convolution

(2 marks)

Q1(b)(i) Edge – intensity discontinuity ∇S_σ (is a maximum) $> T$.

– localized at zero-crossing of $\nabla^2 S_\sigma$ or search for maximum

– used to find contours and describe 2D shape (e.g. SIFT)

– scale, σ, determines low-pass filter cut-off frequency

(3 marks)

(ii) Corner – peak in local autocorrelation fn

– Find $A = \begin{vmatrix} \langle S_x^2 \rangle & \langle S_x S_y \rangle \\ \langle S_y S_x \rangle & \langle S_y^2 \rangle \end{vmatrix}$ where σ_I used in $\langle\,\rangle$ is greater than σ for smoothing

and look for $\det A - k(\mathrm{Trace}\,A)^2 > T$

(4 marks)

(iii) Blob – circular region of uniform intensity

– use band-pass filtering to match to $\sigma^2 \nabla^2 G_\sigma(x,y)$

– localize by max/min of $\sigma^2 \nabla^2 G_\sigma * I(x,y)$

(3 marks)

Q2(a)

(i) Span 3D and large field of view, not - coplanar!
Minimum 6 points. Easy to localise accurately (distinctive?) (2 marks)

(ii) DERIVE

$$\underline{X}_c = R\underline{X}_w + T$$

Rigid body transformation from object to camera 1

Give equations of planar perspective:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{fX_c}{Z_c} \\ \frac{fY_c}{Z_c} \end{bmatrix}$$

Planar perspective projection
f = focal length

no non-linear distortion by lens 2

$$u = u_0 + k_u x$$
$$v = v_0 + k_v y$$

CCD / camera scaling
orthogonal axes 3

Using homogeneous co-ordinates:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \\ 0\,0\,0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} sx \\ sy \\ s \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} \lambda u \\ \lambda v \\ \lambda \end{bmatrix} = \begin{bmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Q2(a)(ii) cont

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} 3 \times 4 \\ p_{ij} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

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

4

$$k = \begin{bmatrix} fk_u & 0 & u_0 \\ 0 & fk_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

(4 marks)

Q2(a) (iii) Each point gives 2 equations in 12 p_{ij}

$$\begin{bmatrix} X_i & Y_i & Z_i & 1 & 0 & 0 & 0 & 0 & -u_iX_i & -u_iY_i & -u_iZ_i & -u_i \\ 0 & 0 & 0 & 0 & X_i & Y_i & Z_i & 1 & -v_iX_i & -v_iY_i & -v_iZ_i & -v_i \end{bmatrix} \begin{bmatrix} p_{11} \\ \vdots \\ p_{34} \end{bmatrix}$$

Solve

$$Ap \simeq 0 \quad \text{by} \quad \lambda_1 \leq \frac{p^T A^T A p}{p^T p} \leq \lambda_{12}$$

Smallest eigenvector of A^TA (corresponding to λ_1).

linear sol^n is optimised by searching for p that minimises the sum of the reprojection errors squared.

$$\min \sum (u_i - \hat{u}_i)^2 + \sum (v_i - \hat{v}_i)^2$$

Q2 a(iv) Rays:

Each u_i and v_i and calibrated camera (known p_{jk}) gives a linear equation in X_i, Y_i, Z_i

Each equation defines a plane
Both planes are <u>not</u> parallel + define / intersect in a ray in space. (2 marks)

Q2(b)(i)

$$\begin{bmatrix} s u_i \\ s v_i \\ s \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{14} \\ p_{21} & p_{22} & p_{24} \\ p_{31} & p_{32} & p_{34} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix}$$

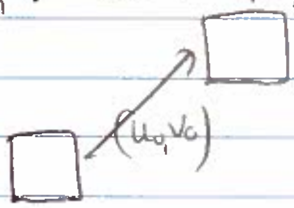
by setting $Z_i = 0$ (above)

$$= \begin{bmatrix} t_{jk} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix}$$

(2 marks)

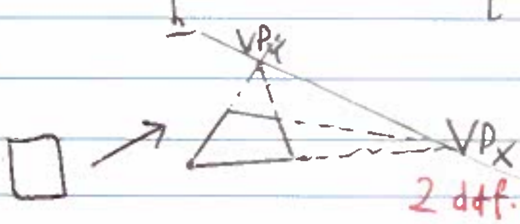
(ii) 8 degrees of freedom: scale (1 dof), rotation (1 dof), shear (2 dof) + translation + foreshortening

Translation $\begin{pmatrix} u_0 \\ v_0 \end{pmatrix}$



$$\begin{bmatrix} 1 & 0 & u_0 \\ 0 & 1 & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$ 2 dof.

perspective (forming)



2 dof.

$$\begin{bmatrix} p_{11} & p_{12} & 0 \\ p_{21} & p_{22} & 0 \\ p_{31} & p_{32} & 1 \end{bmatrix}$$

horizon found from vanishing points (2 dof)

(4 marks)

Q2(b)(iii)

Circle in world $X^2 + Y^2 + f = 0$

Conic in world $aX^2 + bXY + cY^2 + dX + eY + f = 0$

$$[X\ Y\ 1]\begin{bmatrix} a & \frac{b}{2} & \frac{d}{2} \\ \frac{b}{2} & c & \frac{e}{2} \\ \frac{d}{2} & \frac{e}{2} & f \end{bmatrix}\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = 0$$

Check $[X\ Y\ 1]\begin{bmatrix} aX + \frac{b}{2}Y + \frac{d}{2} \\ \frac{b}{2}X + cY + \frac{e}{2} \\ \frac{d}{2}X + \frac{e}{2}Y + f \end{bmatrix} = aX^2 + \frac{b}{2}XY + \frac{d}{2}X + \frac{b}{2}XY + cY^2 + \frac{e}{2}Y + \frac{dX}{2} + \frac{eY}{2} + f$

Can be written as: $\underline{X}^T C \underline{X} = 0$

$\therefore$ Under transformation T

$$\underline{u} = T\underline{X}$$

$$\therefore \underline{X} = T^{-1}\underline{u}$$

$$X^T C X \Rightarrow \underline{u}^T \underbrace{T^{-T} C T^{-1}}_{C'} \underline{u}$$

Circle becomes an ellipse.

(2 marks)

3. (a)

- Using a pre-trained neural network can substantially reduce the quantity of training data required to learn an effective content moderation system. The key intuition here is that we expect that many of the features that are useful for solving the ImageNet classification task are also useful for understanding harmful content.
- There are several valid motivations for truncating the final layer:
  - There are a large number of ImageNet ILSVRC 2012 classes (1000 in total). Consequently, the final fully connected layer would contribute a large number of parameters to the content moderation system, with unclear benefits.
  - The outputs of the final layer are highly specialised to the ImageNet classes, which have relatively limited coverage of human-centric concepts that may be relevant for content moderation (instead, the classes are dominated by animal breeds). By contrast, the penultimate layer is likely to retain less specialised/more general features that may be easier to adapt for the content moderation task.

(b) i.

$$\begin{aligned} s_i &= \frac{\exp(y_i)}{\sum_k \exp(y_k)} \\ \frac{\partial s_i}{\partial s_j} &= \frac{\delta_{ij}\exp(y_i)\Big(\sum_k \exp(y_k)\Big) - \exp(y_i)\exp(y_j)}{\Big(\sum_k \exp(y_k)\Big)^2} \\ &= s_i(\delta_{ij} - s_j) \end{aligned}$$

where $\delta_{ij} = 1$ if $i = j$ and 0 otherwise.

ii.

$$\begin{aligned} L &= H(\mathbf{t}, P(c|\mathbf{x})) \\ L &= -\sum_i t_i \log s_i \\ \frac{\partial L}{\partial y_j} &= -\sum_i t_i \frac{\partial \log s_i}{\partial y_j} \\ &= -\sum_i t_i \frac{1}{s_i}\frac{\partial s_i}{\partial y_j} && \text{by chain rule} \\ &= -\sum_i t_i \frac{1}{s_i} \cdot s_i(\delta_{ij} - s_j) && \text{use softmax derivative from previous question} \\ &= -t_j(1 - s_j) - \sum_{i \neq j} t_i \frac{1}{s_i}(-s_i s_j) \\ &= -t_j + t_j s_j + s_j \sum_{i \neq j} t_i \\ &= -t_j + s_j\Big(\sum_i t_i\Big) && \text{Note: one-hot vector sums to one} \\ \frac{\partial L}{\partial y_j} &= s_j - t_j \end{aligned}$$

(c)

- First step: partition the data into training, validation and test splits. A broad range of options are viable here. A natural default is to shuffle the data, use 60% for training, 20% for validation and 20% for testing (estimating task performance).
- The validation set will be used exclusively for hyperparameter selection.
- A simple but effective optimisation algorithm covered in the lectures is SGD with momentum. Divide the training set into minibatches of data. Let τ denote the current training step. Then compute the parameter update in two steps (this follows the notation in the lectures):

$$\nabla \mathbf{W}^{\tau} = \frac{\partial L}{\partial \mathbf{W}^{\tau-1}} + \epsilon \nabla \mathbf{W}^{\tau-1}$$
$$\mathbf{W}^{\tau} \leftarrow \mathbf{W}^{\tau-1} - \eta \nabla \mathbf{W}^{\tau}$$

  where ϵ is the momentum hyperparameter and η is the learning rate. Another reasonable optimiser choice would be Adam.
- To select hyperparameters (for example, a schedule for the learning rate η), we train models with different hyperparameter settings and compare their performance on the validation set.
- Possible regularisation strategy: insert dropout prior the fully-connected layer.
- Other creative answers could include:
  - Checking the data for annotation errors
  - Checking the data for inappropriate bias
  - Use data augmentation to increase the effective training dataset size (random flips and crops may both be appropriate for content moderation images).
  - After hyperparameter selection is complete, it is possible to merge the training set and validation set and retrain on the resulting set, keeping the hyperparameters fixed.
- Finally, we perform a single evaluation on the test set to estimate task performance.

(d) CNNs are well suited for image processing tasks thanks to several useful inductive biases:

- "translation invariance (/equivariance)" - changes in the input are reflected in the output thanks to the parameter sharing in convolutional kernels. This contributes to the statistical efficiency of CNNs (fewer data samples are required to learn the parameters).
- "locality/sparse connectivity" - convolutional kernels are smaller than the input image/feature map. This means that outputs are only affected by nearby neighbours in the input. This dramatically reduces the number of parameters to be learned and is a good fit for vision tasks (since for the most part, local information is by far the most relevant).

The ResNet architecture in particular makes use of:

- *residual connections* - these assist with optimisation and make it possible to learn deep neural networks more efficiently/robustly.
- *batch normalisation* - this also contributes to improved optimisation.

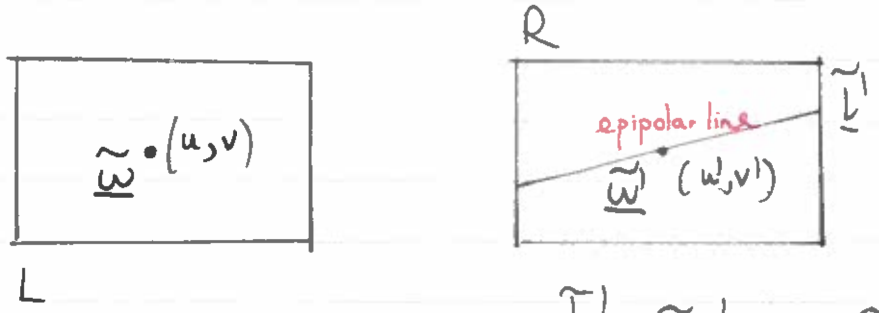
(e) Vision transformers represent a more "data-driven" architecture that no longer gains the statistical efficiency benefits of enforcing translation equivariance and locality in the network. Instead, these models rely on leveraging extremely large pretraining datasets to ensure that the model learns to compensate for the reduced statistical efficiency. In practical terms, Vision Transformers could be deployed by Company Y with one of the following strategies:

- They adopt a feature extractor that has been pretrained on a large labelled dataset (for instance, JFT-300M, rather than ImageNet (ILSVRC 2012)).
- Self-supervised learning is used on a large dataset to pretrain the Vision Transformer (this helps avoid the need for labels).

- They gather a much larger labelled dataset to directly train the model.
- Semi-supervised learning (employing labelled and unlabelled data) is used to fully train the Transformer.

Q4 Definition

(a) (i) The epipolar constraint: pt (u,v) in left view is constrained to lie on line $\underline{\tilde{l}}' = F\underline{\tilde{\omega}}$ in right view.



$$\underline{\tilde{l}}' \cdot \underline{\tilde{\omega}}' = 0$$
$$\underline{\tilde{\omega}}'^T \underline{\tilde{l}} = 0$$
$$\underline{\tilde{l}}' = F\underline{\tilde{\omega}}$$

(1)

(ii) Derivation

$\underline{X}' = R\underline{X} + \underline{T}$ rigid body motion between views

$\underline{X}'$, $\underline{T}$ and $R\underline{X}$ are coplanar due to triangulation

$$\underline{X}' \cdot T_x R \underline{X} = 0$$

$$\underline{X}'^T E \underline{X} = 0 \qquad \text{where } E = [T_x][R]$$

$$T_x = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix}$$

We can express in terms of rays $\underline{p}$ and $\underline{p}'$ where $\underline{p} \parallel \underline{X}$ and $\underline{p}' \parallel \underline{X}'$

(3 marks)

$$\underline{p}'^T E \underline{p} = 0 \quad \text{or} \quad \underline{\tilde{\omega}}'^T K'^{-T} E K^{-1} \underline{\tilde{\omega}} = 0$$

Or in terms of $\underline{\tilde{\omega}}$ and $\underline{\tilde{\omega}}'$ where $\underline{\tilde{\omega}} = K\underline{p}$ and $\underline{\tilde{\omega}}' = K'\underline{p}'$

$\therefore \; F = K'^{-T}[T_\times][R]K^{-1}$ and rank $F = 2$
3x3 matrix
det $F = 0$

7 ddf only

Epipoles $\underline{\tilde{w}_e}$, epipole in left image $\quad F\underline{\tilde{w}_e} = 0$ (1 mark)
$\underline{\tilde{w}_e'}$, epipole in right image $\quad F^T\underline{w_e'} = 0$

Q4(a)(ii) – In stereo vision F is known and for each point (u_i, v_i) we use epipolar constraint to constrain search for correspondence to a line

$$\underline{\tilde{l}_i'} = F\,\underline{\tilde{\omega}_i} \quad \text{and} \quad \underline{\tilde{\omega}_i'} \cdot \underline{\tilde{l}'} = 0$$

– In SFM, F is unknown but we can track features to get correspondences $(u_i\, v_i)$ and (u_i', v_i')

Each pair of correspondences gives the 1D constraint on F

$$[u' \; v' \; 1]\,[F]\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = 0$$

and we $n \geq 8$ correspondences we can estimate F and recover camera motion.

(3 marks)

Q4(b)

(i) Possible keypoint descriptors for features in each image

$\underline{x}_d$

— 128 D vector, SIFT descriptor (unit length)

— 256 D vector, NCC, 16×16 patch of pixels normalised by subtracting mean, divide by $\sqrt{\text{variance}}$.

Correspondence

— Compare euclidean distance between $\underline{x}_d$ and $\underline{x}_k$ in database of descriptors

— Distance between $\underline{x}_d$ and $\underline{x}_k$ $= \sum_{i=1}^{D=128} \left(x_{d_i} - x_{k_i}\right)^2$

— Shortest distance corresponds to best match (NN)

— Accept as a match if satisfies ratio test (look at nearest neighbour $\underline{x}_{k_1}$ and second nearest neighbour $\underline{x}_{k_2}$)

$$\frac{\|x_d - x_{k_1}\|}{\|x_d - x_{k_2}\|} < 0.7$$

(4 marks)

— Best to structure database by k-d tree for efficient search for NN.

Q4(b)

(ii) RANSAC — random chose 8 correspondences and solve for F (see below)

— accept solⁿ with large number of inliers

Solve for F by using $n \geq 8$ correspondences using

$$\begin{bmatrix} u_i' & v_i' & 1 \end{bmatrix} \begin{bmatrix} f_{ij} & & \\ & & \\ & & \end{bmatrix} \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = 0$$

$$\underset{n \times 9}{\begin{bmatrix} \vdots & & & & & & & & \\ u_i' u_i & u_i' v_i & u_i' & v_i' u_i & v_i' v_i & v_i' & u_i & v_i & 1 \\ \vdots & & & & & & & & \\ u_n' u_n & u_n' v_n & u_n' & v_n' u_n & v_n' v_n & v_n' & u_n & v_n & 1 \end{bmatrix}} \underset{\underline{f}}{\overset{9 \times 1}{\begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix}}} = 0$$

Solve by looking for smallest λ_1 and corresponding eigenvector

$$\lambda_1 < \frac{\underline{f}^T A^T A \underline{f}}{\underline{f}^T \underline{f}} < \lambda_9$$

(4 marks)

BUT $\det F = 0$ is not satisfied by the linear solⁿ.

Project F to nearest $\hat{F}$ matrix with $\det \hat{F} = 0$

Q4(b)(iii).

Having estimated F from (b)(ii) we need to decompose it into 2 projection matrices $P_1 = K[I|0]$ and $P_2 = K[R|T]$

— need knowledge of K so that $E = K^T F K = U \Lambda V^T$ and decompose E into R and $\hat{T}$

$$\hat{T}_x = U \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} U^T \quad \text{and} \quad R = U \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} V^T$$

2

Ambiguity in sign of $\hat{T}$ and R vs R^T resolved by making sure reconstructed pts are in front of camera

— need knowledge of $\|T\|$ to give correct scale (from a known reference point or IMU to give baseline distance)

1

— From 2 projection matrices we can solve to recover 3D (X_i, Y_i, Z_i) by <u>triangulation</u>

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = P_1 \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} u_i' \\ v_i' \\ 1 \end{bmatrix} = P_2 \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

we rewrite as 4 equations (linear) in X_i, Y_i, Z_i and solve by least squares

1

$$A \underline{\hat{X}} = 0.$$

(4 marks)