RC 1

4F12 Computer Vision (2019)

Q1 (a) Smoothing to remove high frequency noise which is amplified
(i) by differentiation:

$$\frac{dI}{dx} \xrightarrow{F.T.} j\omega I(j\omega)$$

amplification of high spatial frequencies (2)

(ii) $$g_\sigma(u) \times g_\sigma(v) = G_\sigma(u,v) = \frac{1}{2\pi\sigma} e^{-\frac{(u^2+v^2)}{2\sigma^2}}$$

2D gaussian

$$\therefore S(x,y) = \sum_{-n}^{n}\sum_{-n}^{n} G_\sigma(u,v)\, I(x-u, y-v)$$

Advantage of 1D scheme vs 2D scheme (efficiency)

$2N$ vs N^2 $\left(\text{or } \frac{2N}{N^2}\right)$ operations where $N = 2n+1$ (size of kernel)

(2)

1(b)(i)

$$S(x+\Delta x, y+\Delta y) \simeq S(x,y) + \frac{\partial S}{\partial x}\Delta x + \frac{\partial S}{\partial y}\Delta y$$

$$+ \frac{\Delta x^2}{2}\frac{\partial^2 S}{\partial x^2} + \frac{\Delta y^2}{2}\frac{\partial}{\partial}$$

$$+ \Delta x \Delta y \frac{\partial^2}{\partial x}$$

$\dots$

Consider $\Delta x = 1$, $\Delta y = 1$

$$\therefore S(x+1, y) = S(x,y) + \frac{\partial S}{\partial x} + \frac{1}{2}\frac{\partial^2 S}{\partial x^2} \dots$$

$$S(x-1, y) = S(x,y) - \frac{\partial S}{\partial x} + \frac{1}{2}\frac{\partial^2 S}{\partial x^2}$$

$$\therefore \frac{\partial^2 S}{\partial x^2} \simeq \frac{S(x+1,y) - 2S(x,y) + S(x-1,y)}{1}$$

(2)

Similarly $$\frac{\partial^2 S}{\partial y^2} \simeq S(x, y+1) - 2S(x,y) + S(x, y-1)$$

Kernel

$$\therefore \frac{\partial^2 S}{\partial x^2} : \quad h(u) = \begin{bmatrix} +1 & -2 & +1 \end{bmatrix} \qquad \frac{\partial^2 S}{\partial x^2} = S''_{xx} = \sum_{-1}^{1} h(u)\, S(x-u$$

$$\frac{\partial^2 S}{\partial y^2} : \quad h(v) = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} \qquad \frac{\partial^2 S}{\partial y^2} = S''_{yy} = \sum_{-1}^{1} h(v)\, S(x, y-$$

(2)

(ii) $$\nabla^2 S = \frac{\partial^2 S}{\partial x^2} + \frac{\partial^2 S}{\partial y^2} \qquad h(u,v) = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

(2)

Q1 (c) Image pyramid formed from $S(x, y, \sigma_i)$ where $S(x,y,\sigma_i) = I(x,y) * G_\sigma$

(i). $\nabla^2 S(x,y) = \nabla^2 \, G_\sigma(x,y) * I(x,y)$

$$\simeq (G_{\sigma_{i+1}} - G_{\sigma_i}) * I(x,y)$$

$$\simeq \underline{S(x,y,\sigma_{i+1}) - S(x,y,\sigma_i)} \quad (1)$$

if $\sigma_{i+1} \simeq \underline{1.2\text{-}1.6\sigma_i}$ range

i.e. Laplacian can be approximated by D.O.G operator
Difference of smoothed images in image pyramid if s chosen correctly (1)

Compute $S(x,y,\sigma_i) = G_{\sigma_i} * I(x,y)$
for
– $\sigma_i = \sigma_0 \, 2^{i/s}$ Need $\underline{s = 2 \text{ or } 3 \text{ or } 4}$, s images per octave

– incremental blur $\sigma_K = \sigma_i \sqrt{2^{\frac{2}{s}} - 1}$

– subsample after σ doubles $i = s, 2s \ldots\ldots$
($\frac{1}{4}$ image size)

– $\nabla^2 S(x,y,\sigma_i) \simeq S(x,y,\sigma_{i+1}) - S(x,y,\sigma_i)$ (1)

Look for max/min of $\nabla^2 S$ by evaluating $\underline{26 \text{ neighbours}}$ in scale-space differences. Gives position (x,y) and scale, σ_i.

(1)

(ii) Look for max/min of $\nabla^2 S$ in LOG images. $\underset{(x,y)}{\underline{\text{Position}}}$ and scale, σ_i

Sample 16x16 pixels at $S(x,y,\sigma_i)$. Compute ∇S for each pixel
Histogram (use interpolation) to 10° bins and smooth with gaussian.
Find max of histogram, θ_i = $\underline{\text{dominant orientation}}$ (2)

Q1(c)(iii)

SIFT — 16x16 patch is normalised to scale (size) and orientation (2D) by blob detection / orientation histogram (2D viewpoint change)

— use gradients, $\nabla S(x, y, \sigma_i)$ to encode 2D shape (robust to lighting changes)

— normalize 128D vector to unit length and truncate any values > 0.2 (invariance to lighting and speculaties)

— Local histograms of gradients give invariance to small viewpoint changes and occlusion.

(3)

Limitations — fail at occluding boundaries (strong edges)

— can not cope with matches under large viewpoint changes (es > 30°) from perspective

Q2(a)(i).

Algebraic — homogeneous co-ordinates

$$\underset{3D}{\overset{\text{homogeneous}}{\begin{pmatrix} su \\ sv \\ s \end{pmatrix}}} \simeq \underset{2D}{\overset{\text{image plane}}{\begin{pmatrix} u \\ v \end{pmatrix}}}$$

$$\underline{\tilde{\omega}} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \text{ where } S = x_3 \quad \underline{\omega} = \begin{pmatrix} x_1/x_3 \\ x_2/x_3 \end{pmatrix}$$

Geometric — (inverse) depth scaling (distance along optical axis)

$$s = p_{31}X + p_{32}Y + p_{33}Z + p_{34} = Z_{camera}$$

(2)

(ii)

$$\overset{3\times1}{\begin{bmatrix} su \\ sv \\ s \end{bmatrix}} = \overset{3\times3}{\begin{bmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}} \overset{3\times4}{\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}} \overset{4\times4}{\begin{bmatrix} R & T \\ 000 & 1 \end{bmatrix}} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

$\underline{\tilde{\omega}}$ 4 dof 1 6 dof $\tilde{X}$

$$\tilde{\omega} = \overset{3\times4}{P} \underline{\tilde{X}}$$

11 parameters of 10 d.o.f. since $\alpha_u = fk_u$, $\alpha_v = fk_v$

(4)

(iii)

$$\begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix}^T = (KR)^T = R^T K^T$$

Decompose by QR to give

$Q = R^T$ (orthonormal) and $R = K^T$ (~~lower tri~~ upper triangle) (1)

$$\underline{T} = K^{-1} \begin{bmatrix} p_{14} \\ p_{24} \\ p_{34} \end{bmatrix}$$

(1).

(i.F17

Q2 (b) Look at $Z=0$, X–Y plane.

(i)

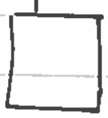
$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{14} \\ p_{21} & p_{22} & p_{24} \\ p_{31} & p_{32} & p_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

$3\times1 \qquad 3\times3 \qquad 3\times1$

9 parameters up to arbitrary scale $\therefore$ 8 d.o.f. (2)

(ii)

world square.



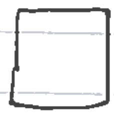
$\begin{bmatrix} \cdot & \cdot & x \\ \cdot & \cdot & x \\ \cdot & \cdot & 1 \end{bmatrix}$ translation 2 dof

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

$\begin{bmatrix} x & x & \cdot \\ x & x & \cdot \\ \cdot & \cdot & 1 \end{bmatrix}$ rotation, 1 dof,

θ

$\begin{bmatrix} x & \cdot & \cdot \\ \cdot & x & \cdot \\ \cdot & \cdot & 1 \end{bmatrix}$ scale, 1 dof, Δ



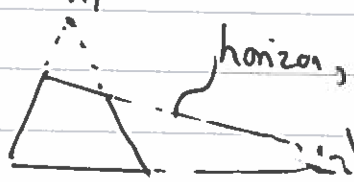
area ↑ by Δ^2

$\begin{bmatrix} x & x & \cdot \\ x & x & \cdot \\ \cdot & \cdot & 1 \end{bmatrix}$ pure shear, 2 dof

axis and magnitude.

perspective change,

$\begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ x & x & 1 \end{bmatrix}$ fanning 2 dof.



(2)

7

Q2(b)(iii)

Consider lines // to X at ∞

$$\begin{array}{cc} & X=\infty \\ \underline{V}_{px} = \begin{bmatrix} & p_{ij} & \end{bmatrix} & \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} p_{11} \\ p_{12} \\ p_{13} \end{bmatrix} \end{array}$$

Consider // lines to Y axis at ∞

$$\begin{array}{cc} & Y=\infty \\ \underline{V}_{pY} = \begin{bmatrix} & p_{ij} & \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} p_{21} \\ p_{22} \\ p_{23} \end{bmatrix} \end{array}$$

2

These points lie on horizon, $\underline{l}$, such that $\underline{l} \cdot \underline{\omega} = 0$

Horizon $\underline{l}$ must satisfy: $\underline{l} \cdot \underline{V}_{px} = 0$ and $\underline{l} \cdot \underline{V}_{pY} = 0$

$$\therefore \underline{l} = \begin{vmatrix} \underline{i} & \underline{j} & \underline{k} \\ p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \end{vmatrix}$$

$$\therefore \quad \underline{l} = \begin{pmatrix} p_{12}p_{23} - p_{13}p_{22} \\ p_{13}p_{21} - p_{11}p_{23} \\ p_{11}p_{22} - p_{21}p_{12} \end{pmatrix}$$

2

line in image (horizon)

$$\therefore \left(p_{12}p_{23} - p_{13}p_{22}\right)u + \left(p_{13}p_{21} - p_{11}p_{23}\right)v + \left(p_{11}p_{22} - p_{21}p_{12}\right) = 0$$

Q2(c) AR need to recover 3D projection matrix in <u>real-time</u>

- $\underline{w} = \left[\underset{3\times 4}{\underline{p}} \right] \left[\tilde{\underline{X}} \right]$

- Can calibrate from known 3D points $(\underline{X})$ and their correspondence $(\underline{w})$.
  $N \geq 6$ for 3D or $N \geq 4$ for points on the plane
  $A\underline{p} = 0$

- Best to assume known K, internal parameters, which are fixed for mobile phone type

- Can use standard ARtool kit <u>marker</u> for calibration (eg Zappar code)

- <u>Fast</u> solution needed. Find "corners" or FAST features in image and track by normalized correlation. Use plane features

- Estimate homography $H = K[\underline{r}_1\ \underline{r}_2\ \underline{t}]$ and $\underline{r}_3 = \underline{r}_1 \times \underline{r}_2$
  decompose to get new pose $\underline{t}$ and $\underline{R}$.
  (approx orientation is given by IMU and gyroscopes)

- <u>Render CG using projection matrix</u>, $K[R|t]$.

(4)

Q3

(a)

(i)

$$\underline{X}' = R\underline{X} + \underline{T}$$

The rays, $\underline{X}'$, $\underline{X}$ and $\underline{T}$ are co-planar.

$\therefore$ taking cross product with $\underline{T}$ and dot product with $\underline{X}'$ $\Rightarrow$ triple scalar prod $= 0$

$$\underline{X}' \cdot \underbrace{[\underline{T}_\times R]}_{E} \underline{X} = 0$$

$$E \text{ if } \underline{T}_\times R = [T_\times] R$$

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

skew-symmetric matrix

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

$$\therefore \quad \underline{p}'^T E \, \underline{p} = 0$$

In terms of image cordinates: $\underline{w}' = K'\underline{p}'$ and $\underline{w} = K\underline{p}$

$\therefore$

$$\underline{w}'^T K'^{-T} E K^{-1} \underline{w} = 0$$

$$\underline{w}'^T \underbrace{[K'^{-T}][T_\times][R][K^{-1}]}_{F} \underline{w} = 0$$

4 matrices

(5)

Q3(a)(ii)

Epipolar constraint

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

- Points on right view must lie on epipolar line $\underline{l}' . \underline{\omega}' = 0$

or $\underline{\omega}'^T \underline{l} = 0$

$$\therefore \underline{l}' = F \underline{\omega}$$

ie. line corresponding to image point (left) $\underline{\omega}$

- Epipole in right view, $\underline{e}'$ lies on all epipolar lines $\underline{l}'$

$$\therefore \underline{l}' . \underline{e}' = 0 \text{ for all } \underline{l}' = F \underline{\omega}$$

$$\therefore \omega^T F^T \underline{e}' = 0$$

$$\underline{F^T \underline{e}' = 0} \quad \text{null-space of } F^T$$

(3)

Q3(b)

(i) Matching (corresponding) points:

For each image:

- Find interest points such as blobs (scale, position and orientation)
- SIFT descriptor. (128D vector)

- Match in right view by NN and k-d search tree:
(Compute dot. product)

[Alternative is to find Harris corner and match by normalized cross-correlation]

(2)

(ii) Need 8 pt correspondences with a linear method or 7 correspondences if wish to include Rank F = 2, ie det F = 2.

Solve $Af = 0$ by minimising such that $|f| = 1$ (2) $\lambda_1 \leq \frac{f^T A^T A f}{f^T f} \leq \lambda_8$

(iii) Least-squares followed by rank F = 2 by projection or SVD $\lambda_3 = 0$.
RANSAC to remove incorrect matches.

(2)

Non-linear optimisation of 7 parameters to minimize the algebraic error.

Q3(c)(i) Estimate F

Compute $E = K'^T F K = T_x R = U \Lambda V^T$ (SVD decom

$$T_x = U \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} U^T \qquad R = U \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} V^T$$

- skew-symmetric
- unknown scale $|T|$?

- orthonormal matrix

This decomposition gives 4 solutions $\pm T$, R and R^T

Check by computing depths, must be positive.

Left with unknown scale $|T|$. Need a known length in world.

(4)

$$P_L = K[I|0] \quad \text{and} \quad P_R = K'[R|T]$$

(c)(ii) We have $w' = K'(R\ T]\,X$ and $w = K[I|0]X$

Find X by triangulation

i.e. by looking at stereo correspondence w' and w

Each point (visible) gives 2 equations in 3 unknowns $(X, Y, Z,$

(i.e. 2 planes defining a ray)

$\therefore$ 4 equations in X, Y, Z. Solve by least-squares

(2)

RET(13

a) A complete answer will include 4 of the following points

Stage 1 : 1) compute high level feature description of both images

2) both images passed through the same CNN so there are fewer parameters to learn & so ordering of input images do not matter

Stage 2 : 3) compute similarity of the two images via the weighted squared distance and add a bias.

4) Again order of inputs does not matter.

5) Bias required so that when two identical images are passed in, the activation will be large & positive (resulting in $p(y=1|z_1,z_2) \approx 1$

Stage 3 : 6) converts real valued activations that lie in $(+\infty, -\infty)$ to probabilities that lie between $(0,1)$.

b) This & the next question are bookwork in disguise being identical to logistic regression covered in lectures.

$$\mathcal{L}(\phi, w) = \log \prod_n p(y^{(n)} \mid z_1^{(n)}, z_2^{(n)}, \underline{w})$$
$$= \sum_n \left[y_n \log \sigma(a^{(n)}) + (1-y_n) \log(1-\sigma(a^{(n)})) \right]$$

c)
$$\frac{d\mathcal{L}(\phi,w)}{dw_d} = \sum_n \frac{d\mathcal{L}}{da^{(n)}} \frac{da^{(n)}}{dw_d} \qquad \text{(chain rule)}$$

$$\frac{d\mathcal{L}}{da^{(n)}} = \frac{y_n}{\sigma(a^{(n)})} \frac{d\sigma(a^{(n)})}{da^{(n)}} - \frac{(1-y_n)}{1-\sigma(a^{(n)})} \frac{d\sigma(a^{(n)})}{da^{(n)}}$$
$$= \frac{y_n(1-\sigma(a^{(n)})) - (1-y_n)\sigma(a^{(n)})}{\sigma(a^{(n)})(1-a^{(n)})} \frac{d\sigma(a^{(n)})}{da^{(n)}}$$
$$= \frac{y_n - \sigma(a^{(n)})}{\sigma(a^{(n)})(1-\sigma(a^{(n)}))} \frac{d\sigma(a^{(n)})}{da^{(n)}}$$

cancel

$$\frac{d\sigma(a^{(n)})}{da^{(n)}} = \frac{d}{da^{(n)}} \frac{1}{1+e^{-a^{(n)}}} = -1 \times \left(\frac{1}{1+e^{-a^{(n)}}}\right)^2 \times \left(-e^{-a^{(n)}}\right)$$
$$= \sigma(a^{(n)})^2 \left(\frac{1}{\sigma(a^{(n)})} - 1\right)$$
$$= \sigma(a^{(n)})(1-\sigma(a^{(n)}))$$

$$\frac{da^{(n)}}{dw_d} = \begin{cases} 1 & \text{if } d=0 \\ h_{1d}-h_{2d} & \text{if } d>0 \end{cases}$$

$$\therefore \boxed{\frac{d\mathcal{L}}{dw_d} = -\sum_n \left[\sigma(a^{(n)}) - y^{(n)}\right]\left[\delta_{d0} + (1-\delta_{d0})[h_{1d}-h_{2d}]\right]}$$

d) Use gradient ascent of log likelihood:

initialise $\underline{w}$ & then use:

$$\underline{W}^{(new)} = \underline{w}^{(old)} + \eta \frac{d\hat{\mathcal{L}}}{d\underline{w}}$$

η = learning rate

NB. move in direction of gradient here as maximising

$\frac{d\hat{\mathcal{L}}}{d\underline{w}}$ = estimator for gradient produced by selecting M training data points at random @ each iteration

$$= -\frac{N}{M} \sum_m \left[\sigma(a^{(m)}) - y^{(m)}\right]\left[\delta_{d0} - (1-\delta_{d0})[h_{1d}-h_{2d}]\right]$$

Can also discuss: initialisation, adapting the learning rate, momentum etc.

4F12

Module 4F12 (Computer Vision) Assessor's Comments

1. **Gaussian smoothing, bandpass filtering and SIFT**. Attempted by 69/77 Part IIB candidates, average mark 13.5/20.

   The first part was a very straightforward question covering convolution with low pass filters and was well answered by most candidates. The second and third parts were more challenging. There were very few correct attempts at obtaining the laplacian kernel in (b). Candidates also struggled to describe how SIFT achieved invariance to lighting and viewpoint and how it was encoding 2D shape in (c).

2. **Perspective projection and camera calibration**. Attempted by 64/77 candidates, average mark 14.2/20.

   A question covering perspective projection. Well answered by most candidates. A few marks were lost on the recovery of the horizon (b). The only challenging component was the application of the theory in (a) and (b) to a practical example of calibration for AR.

3. **Epipolar geometry and stereo vision**. Attempted by 58/77 candidates, average mark 14.3/20.

   The derivation in (a) proved to be very easy for most candidates. Marks were lost in (b) for insufficient details in how to findi point correspondences using features and descriptors and the estimation of the fundamental matrix in presence of outliers and noise (b). Part (c) was found to be easy for most candidates.

4. **Object detection with convolutional neural networks**. Attempted by 40/77 candidates, average mark 14/20.

   Many candidates that attempted this question made excellent progress.