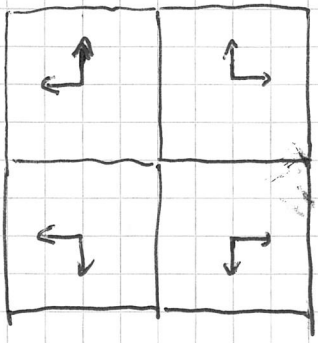
EXAM JUNE 5TH 2018

1]

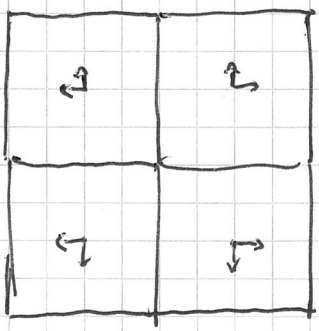
a)

Vector bouquet: (8 bins):

Vector bouquets for P (some order of cells):



Vector bouquets for Q:



There are fewer pixels with gradients, so the entries in the gradient histogram are smaller for Q than for P

There are 4 cells, each with 8 histogram bins

→ $4 \cdot 8 = 32$ dimensions

b)

	distances to 1st neighbor	distances to 2nd neighbor	ratio
d_1	0.1 (q_1)	0.2 (q_2)	0.5 = 1/2
d_2	0.1 (q_2)	0.4 (q_1)	0.25 = 1/4
d_3	2 (q_3)	8 (q_2)	0.25 = 1/4
d_4	2 (q_3)	12 (q_2)	1/6

According to the Lowe ratio, the match (d_4, q_3) is most certain.

c) The density of the SIFT descriptor space is non-uniform. There are regions, e.g., corresponding to repeating structures, that are denser. Thus, absolute distances are unreliable. ~~to~~ Rather, we need to normalize the distances.

2 ||

a) Loss for training example I_i with true label 5:

$$L(I_i) = -\ln(p_5(I_i))$$

b) For simplicity: $p_5(I_i) = p_5$, $p_j(I_i) = p_j$

Softmax function:

$$p_5 = \frac{e^{y_5}}{\sum_i e^{y_i}}$$

y_i are outputs of fully connected layer

We want to compute the gradient

$$\frac{\partial L(I_i)}{\partial y_5}$$

Using the chain rule, we get

$$\frac{\partial L(I_i)}{\partial y_5} = \frac{\partial L(I_i)}{\partial p_5} \frac{\partial p_5}{\partial y_5}$$

Where $\frac{\partial L(I_i)}{\partial p_5} = -\frac{1}{p_5}$

We use the quotient rule $\frac{f'(x)g(x) - f(x)g'(x)}{(g(x))^2}$

to compute $\frac{\partial p_5}{\partial y_5}$:

~~$\frac{\partial p_5}{\partial y_5}$~~ ~~Here $f(x) = e$~~

Here: $f(y_5) = e^{y_5}$, $f'(y_5) = e^{y_5}$

$g(y_5) = \sum_i e^{y_i}$, $g'(y_5) = e^{y_5}$

this gives us:

$$\frac{\partial p_5}{\partial y_5} = \frac{e^{y_5} g(y_5) - e^{y_5} e^{y_5}}{(g(y_5))^2}$$

$$= \frac{e^{y_5}}{g(y_5)} \cdot \frac{g(y_5) - e^{y_5}}{g(y_5)}$$

2b Continued

$$= \frac{e^{y_5}}{\sum_i e^{y_i}} \cdot \frac{\sum_i e^{y_i} - e^{y_5}}{\sum_i e^{y_i}} = p_5 (1 - p_5)$$

$$\Rightarrow \frac{\partial L(I_i)}{\partial y_5} = -\frac{1}{p_5} p_5 (1 - p_5) = p_5 - 1$$

c) Input dimensions:

$46 \times 46 \times 3$ (original input image) $\rightarrow$ $44 \times 44 \times 10$ (no padding) $\rightarrow$ $44 \times 44 \times 10$ (after ReLU)

$\rightarrow$ $22 \times 22 \times 10$ (after max. pooling) $\rightarrow$ $20 \times 20 \times 20$ (due to no padding)

$\rightarrow$ $20 \times 20 \times 20$ (after ReLU) $\rightarrow$ $10 \times 10 \times 20$ (after max. pooling)

$\rightarrow$ $10 \rightarrow 10$

Number of parameters: (for the bias term)

$\times$ 10 3×3 convolutions: $10 \cdot (3 \cdot 3 \cdot 3 + 1) = 280$

$\times$ 20 3×3 convolutions: $20 \cdot (10 \cdot 3 \cdot 3 + 1) = 1820$

$\times$ Fully connected layer:

input size: $10 \times 10 \times 20 = 2000$

output size: 10

$\Rightarrow$ $2000 \times 10 = 20000$ parameters

d) Replace the FC layer by 10 10×10 convolutions. This will result in exactly the same output for an input image of size 46×46.

3|

a) Re-write the system as a function of the parameters of the affine transform:

$$\begin{pmatrix} \tilde{x} & \tilde{y} & \tilde{z} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \tilde{x} & \tilde{y} & \tilde{z} & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \tilde{x} & \tilde{y} & \tilde{z} & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \\ i \\ t_x \\ t_y \\ t_z \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

One match gives 3 equations in 12 unknowns → we need 4 matches.

Stack the systems into a system $A\theta = b$, where $A \in \mathbb{R}^{12\times 12}$, $b \in \mathbb{R}^{12}$ and solve in Matlab as $\theta = A \backslash b$.

b) The minimal solver will create a solution such that the K points used to compute it are inliers. Thus, we will always have at least K inliers. A solution with K inliers is thus not trustworthy

c) In 2D, only 3 matches are needed.

Probability of selecting 3 inliers: $(0.5)^3$

↳ On average $\frac{1}{(0.5)^3}$ samples needed

In 3D: $1/(0.5)^4$ samples needed

↳ Finding a good transformation takes

$$\frac{(0.5)^3}{(0.5)^4} = 2 \text{ times longer.}$$

4|

a) ransac_triangulation ($\{P_i\}, \{u_i\}, t$)

$k_{max} = \lceil \log(0.01) / \log(0.1 - (\frac{2}{n})^2) \rceil$

$I_{max} = 0$;

for $k = 1 : k_{max}$

$P_{s,us}$ = random sample of size 2 from $\{P_i\}, \{u_i\}$

U_{hat} = minimal_solver (P_s, u_s)

reprojection_errors = reprojection-errors ($\{P\}, \{u\}, U_{hat}$)

~~if sum (reprojection~~

I = sum (reprojection_errors $\leq t$)

if $I > I_{max}$

$I_{max} = I$

$U_{best} = U_{hat}$

$\varepsilon = \frac{I}{n}$

$k_{max} = \lceil \log(0.01) / \log(1 - \varepsilon^2) \rceil$

end

end

return U_{best}, I_{max};

b) The projected point in homogeneous coordinates is given as

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = P \begin{pmatrix} U \\ 1 \end{pmatrix}$$

This yields the following image coordinates:

$$\begin{pmatrix} u' \\ v' \end{pmatrix} = \frac{1}{z'} \begin{pmatrix} x' \\ y' \end{pmatrix}$$

This yields the following residuals:

$$r_x = x - u' = x - \frac{1}{z'} x'$$

$$r_y = y - v' = y - \frac{1}{z'} y'$$

The equations are valid for $z' > 0$, i.e., if U is in front of the camera

5|

a) Using the chain rule, we can express $\frac{\partial L_i}{\partial w_n}$ as

$$\frac{\partial L_i}{\partial w_n} = \frac{\partial L_i}{\partial y_n} \frac{\partial y_n}{\partial w_n}$$

Since $y_n = x_n w_n$, we have $\frac{\partial y_n}{\partial w_n} = x_n$

L_i depends on the $\tilde{y}_j$, applying the chain rule, we obtain

$$\frac{\partial L_i}{\partial w_n} = x_n \sum_j \frac{\partial L_i}{\partial \tilde{y}_j} \frac{\partial \tilde{y}_j}{\partial y_n}$$

$\tilde{y}_j$ depends on y_n through z and $\tilde{x}$.

We thus obtain

$$\frac{\partial \tilde{y}_j}{\partial y_n} = \frac{\partial \tilde{y}_j}{\partial \tilde{x}} \frac{\partial \tilde{x}}{\partial z} \frac{\partial z}{\partial y_n}$$

We have $z = \sum_m y_m \Rightarrow \frac{\partial z}{\partial y_n} = 1$

We have $\tilde{x} = \xi(z) = \frac{e^z}{1+e^z}$

Thus $\frac{\partial \tilde{x}}{\partial z} = \frac{e^z(1+e^z) - (e^z)^2}{(1+e^z)^2} = \frac{e^z}{(1+e^z)^2}$

We have $\tilde{y}_j = \tilde{w}_j \tilde{x}$

$$\Rightarrow \frac{\partial \tilde{y}_j}{\partial \tilde{x}} = \tilde{w}_j$$

$$\Rightarrow \frac{\partial L_i}{\partial w_n} = x_n \sum_j \tilde{w}_j \frac{e^z}{(1+e^z)^2} \frac{\partial L_i}{\partial \tilde{y}_j}$$

b) Plugging the values given into the formula from (a) yields

Text Text

$$z = 2 - 4 + 2 = 0 \Rightarrow \frac{e^z}{(1+e^z)^2} = \frac{1}{4}$$

$$\Rightarrow \frac{\partial L_i}{\partial w_1} = \frac{2}{4}(2 \cdot 1 + 4 \cdot 1 + 1 \cdot 2)$$

$$= \frac{1}{2}(2+4+2) = 4$$

c) The next value for w_1 will be

$1 - 0.01 \cdot 4 = 0.96$

6| Equation for perpendicular distance:

$$r(x,y) = \frac{ax + by + c}{\sqrt{a^2+b^2}}$$

or $r(x,y) = x \cos\alpha + y \sin\alpha + c$

Sum of least squares:

$$\sum (r(x,y))^2 = \sum (x_i \cos\alpha + y_i \sin\alpha + c)^2 = f(\alpha, c)$$

The line goes through the mean point:

$$\frac{\partial f(\alpha,c)}{\partial c} = \sum 2 (x_i \cos\alpha + y_i \sin\alpha + c) \stackrel{!}{=} 0$$

$$\Leftrightarrow \quad \cos\alpha \cdot \sum x_i + \sin\alpha \cdot \sum y_i + c \sum_n 1 = 0$$

$$\Leftrightarrow \quad \cos\alpha \frac{\sum x_i}{n} + \sin\alpha \cdot \frac{\sum y_i}{n} + c = 0$$

$\Rightarrow$ work on centered data, ie., shift ~~data~~ center into origin

$\Rightarrow$ new line $x \cos\alpha + y \sin\alpha + c' = 0$ with $c' = 0$

2 ways to solve for the normal $(\cos\alpha, \sin\alpha)$ of the line:

1) Closed form solution:

Let $x_i' = x_i - \mu_x$, $y_i' = y_i - \mu_y$

$$\mu_x = \sum \frac{x_i}{n}, \quad \mu_y = \sum \frac{y_i}{n}$$

* Consider centered line:

$$f(\alpha) = \sum (x_i' \cos\alpha + y_i' \sin\alpha)^2$$

$$\frac{\partial f(\alpha)}{\partial \alpha} = \sum 2 (x_i' \cos\alpha + y_i' \sin\alpha)(y_i' \cos\alpha - x_i' \sin\alpha) \stackrel{!}{=} 0$$

$$\Rightarrow \quad \sum \cos^2\alpha\, x_i' y_i' + y_i'^2 \sin\alpha \cos\alpha - x_i'^2 \cos\alpha \sin\alpha - \sin^2\alpha\, x_i' y_i' \stackrel{!}{=} 0$$

$$\Leftrightarrow \quad \cos^2\alpha \sum x_i' y_i' + \cos\alpha \sin\alpha \sum (y_i'^2 - x_i'^2) - \sin^2\alpha \sum x_i' y_i' = 0$$

$$\Leftrightarrow \quad \underbrace{\sum x_i' y_i'}_{a} (\cos^2\alpha - \sin^2\alpha) + \cos\alpha \sin\alpha \underbrace{\sum (y_i'^2 - x_i'^2)}_{b} = 0$$

$$\Leftrightarrow \quad (\cos^2\alpha - \sin^2\alpha)\, a + b \cos\alpha \sin\alpha = 0$$

6 continued

$$\Rightarrow \quad 0 = \frac{a\cos^2\alpha}{\cos^2\alpha} - \frac{a\sin^2\alpha}{\cos^2\alpha} + b\frac{\cos\alpha\sin\alpha}{\cos^2\alpha}$$

$$\Rightarrow \quad 0 = a - a\tan^2\alpha + b\tan\alpha$$

$\tan\alpha = \frac{\sin\alpha}{\cos\alpha}$

$$\Rightarrow \quad \tan\alpha_1 = \frac{-b+\sqrt{b^2+4a^2}}{-2a} = \frac{b-\sqrt{b^2+4a^2}}{2a}$$

$$\text{or} \quad \tan\alpha_2 = \frac{-b-\sqrt{b^2+4a^2}}{-2a} = \frac{b+\sqrt{b^2+4a^2}}{2a}$$

⇒ select α_i that minimizes errors, then compute c usin α_i

2) Alternative solution:

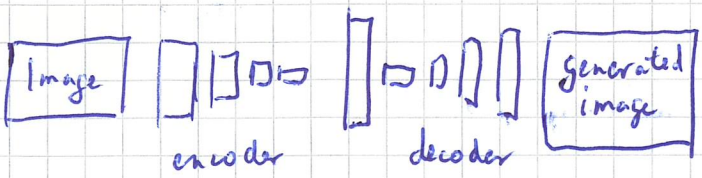
Compute the covariance matrix

$$\Sigma = \sum_k \begin{pmatrix} x_i'^2 & x_i' y_i' \\ x_i' y_i' & y_i'^2 \end{pmatrix}$$

The eigenvector corresponding to the smaller eigenvalue of Σ corresponds to $(\cos\alpha, \sin\alpha)$

Example Question: Variational Autoencoders

a)



Given an image x as input, the encoder (typically a convolutional neural network) maps it to a point z in the latent space. The encoder thus defines a probability distribution $q(z|x)$ in the latent space.

Given a point z in the latent space as input, the decoder (typically a CNN) maps z to an image x. The decoder thus defines a probability distribution $p(x|z)$ in the space of images.

b) Given a simple probability distribution $p(z)$ on the latent space (typically a normal distribution), we sample a point z in latent space according to $p(z)$. We then use the decoder to map it to an image.

c) The term $D_{KL}(q(z|x) \| p(z))$ minimizes the Kullback-Leibler (KL) divergence between the distributions $q(z|x)$ and $p(z)$ defined above. The KL divergence term tries to align $q(z|x)$ (which is learned through the encoder) with the chosen distribution $p(z)$. Intuitively, the term tries to ensure that points z that are likely to be

Sampled ~~according~~ according to $p(z)$ correspond to points that likely result in valid generated images.