UNIVERSITY OF CALIFORNIA LOS ANGELES

Department of Computer Science

Final Exam – Winter 2021

CS 188.2 Introduction to Computer Vision

Start: Monday 3/15/21 8 am PT
Due: Tuesday 3/16/21 8 am PT

First Name: Ziyi

Last Name: Yang

UID: 405064363

This is an open book, open Internet test; however, you must do the exam by yourself. Answer all questions and submit your answers as a PDF to Gradescope. Free-form questions should be answered in 2-5 sentences. DO NOT simply write everything you know about the topic of the question. Answer the question that was asked. Extraneous information not related to the answer to the question will not improve your grade and may make it difficult to determine if the pertinent part of your answer is correct. Write and sign the UCLA code of conduct (given below) on your answer sheets. **Your exam won't be graded if the signed code of conduct is missing from your submission.**

UCLA Code of Conduct:
I agree to the UCLA academic integrity code, that no illicit aid has been used or provided.

Signature: Ziyi Yang 杨子羿

Question	Points	Score
1	5	
2	5	
3	8	
4	3	
5	4	
6	8	
7	10	
8	5	
9	4	
10	8	
11	3	
12	2	
Total:	65	

Basic Probability

1. For parts (a) and (b) consider 2 probability distributions $p_1(x) = \mathcal{N}(\mu_1, \sigma_1)$ and $p_2(x) = \mathcal{N}(\mu_2, \sigma_2)$. $\mathcal{N}(\mu, \sigma)$ refers to a Gaussian distribution with mean μ and standard deviation σ.
   (a) (2 points) Compute KL divergence between p_1 and p_2, i.e. $KL(p_1||p_2)$. Note that no credit will be given if you do not show the derivation.
   (b) (1 point) Compute KL divergence between p_2 and p_1, i.e. $KL(p_2||p_1)$.
   (c) (2 points) KL divergence is not symmetric w.r.t p_1, p_2, i.e. $KL(p_1||p_2) \neq KL(p_2||p_1)$. Use $KL(p_1||p_2)$ and $KL(p_2||p_1)$ to construct a function $f(p_1||p_2)$ which is symmetric w.r.t p_1, p_2.

Classification

2. Consider a binary (0/1) classification setup with binary label y for a given image x. y is hence assumed to follow a Bernoulli distribution w.r.t x; in other words, $p(y|x)$ is assumed to be a Bernoulli distribution. Assume the output of your classifier is $\hat{y} \in (0, 1)$. As seen in problem set 4, you can hence write $p(y|x) = \hat{y}^y(1 - \hat{y})^{(1-y)}$.
   (a) (2 points) Find a, b such that $p(y|x) = ay + b$ and $p(y|x) = \hat{y}^y(1-\hat{y})^{(1-y)}$ are equivalent for $y = 0, 1$. No credit will be awarded if the approach to find a, b is not shown. It is fine to write a, b as functions of $\hat{y}$.
   (b) (1 point) We can hence define a new loss function $L(y, \hat{y}) = -ay - b$ where a, b are the values obtained in the previous part. Compute $\frac{\partial L}{\partial \hat{y}}$.
   (c) (2 points) Even though the two formulations of $p(y|x)$ seem point wise equivalent, can $L(y, \hat{y}) = -p(y|x) = -ay - b$ be used to train a classifier via stochastic gradient descent? [Hint: Use your answer in part (b)]

Gradient Descent

3. Consider a loss function $L(y^*, y) = \beta(y - y^*)^2$. y^* and β are constants with $\beta > 0$. Let y denote the output of a machine learning model. The goal is to minimize this loss using gradient descent and find the optimal value of y.
   (a) (1 point) What is the minimum possible value of $L(y^*, y)$?
   (b) (1 point) Can you derive a closed form expression for the optimal value of y which is a minimizer of $L(y^*, y)$? If yes, derive the closed form expression and the optimal value of y.
   (c) (1 point) For all the sub-parts here onwards, assume $y^* = 0$. Compute $\frac{\partial L}{\partial y}$.
   (d) (1 point) Write the gradient update step for the variable y, assuming a learning rate α.
   (e) (2 points) Assuming the initial value of y to be y_0, what will be the value of y after n updates?
   (f) (2 points) Find a range of values of the learning rate α for which y will converge to its optimal value derived in part (b) as $n \to \infty$.

3-D geometry

4. (3 points) Suppose you apply the 8 point algorithm and obtain the fundamental matrix to be $F = U \operatorname{diag}[3, 5, 0]\ V^T$, where U and V are the matrices in the singular value decomposition of F. Project F to the space of essential matrices to obtain the best approximation to the essential matrix for this system of correspondences.

2D Transformations and Homography

5. (4 points) Show that the origin in 2D coordinates does not necessarily map to the origin under 2D projective transformations.

6. Amidst the pandemic blues, you decide to take a walk to the UCLA campus, and capture a video of Royce Hall. The spatio-temporal description of a point in this video can be given as $I(x, y, t)$, where x, y denote the spatial location in the frame, and t denotes the temporal id of a video frame. Assume the video has N frames, i.e. t ranges from 0 to $N-1$. Since the campus is closed, your video has people and objects moving sparsely across the scene, with the majestic Royce Hall in the background. The camera and the background scene are held fixed w.r.t to each other while this video is captured.
   (a) (3 points) Propose an algorithm to extract an Image of Royce Hall (static background from the video). [Hint: Can you apply one of the filters learned in class to the video frames for this problem? The solution ideally wouldn't be more than 5 lines.]
   (b) (5 points) You posted this video online, motivating your friend to also visit the Royce hall and take a video. But instead of keeping the camera fixed, your friend rotated the camera while taking the video. Will the algorithm in part (a) work for the video captured by your friend? If yes, justify. If no, propose a modification to your answer in part (a) to extract the image of the background. Assume that the only change to the camera pose was rotation (i.e. there was no camera translation).

CNNs

7. Consider a 2D convolution of an image I of size $h \times w$ with a kernel K of size $n \times n$. Let the output of this convolution be denoted $I \circledast K$, which has size $r \times c$. Assume the convolution has no padding.
   (a) (4 points) Assume we can flatten the image I and the convolution output $I \circledast K$ into column vectors $I', (I \circledast K)'$. Prove that there exists a matrix K' such that the convolution operation can be evaluated in terms of matrix multiplication, i.e. $(I \circledast K)' = K'I'$.
   (b) (3 points) Propose a method using $K^{'T}$ to transform an input of size $r \times c$ into an output of size $h \times w$. This idea is the key idea behind transposed convolutions, which are used for obtaining pixel-wise predictions from CNNs.
   (c) (3 points) Assume we compute the 2D convolution using matrix multiplication, i.e. $(I \circledast K)' = K'I'$. Does there always exist a transposed convolution that inverts this 2D convolution? In other words, for any kernel K with associated matrix K' (defined in part a), does there always exist a matrix $K_{inv}^{'}$ such that $I' = K_{inv}^{'}((I \circledast K)')$? Please justify your answer.

8. (5 points) Consider the following 3 layer fully-convolutional neural network: CONV1-MAXPOOL-CONV2-MAXPOOL-CONV3. Here

   - CONV1 is a 2-d convolutional layer with input-channels=3, output-channels=10, filter-size=5, stride=2, padding=0 (no padding)
   - CONV2 is a 2-d convolutional layer with input-channels=10, output-channels=20, filter-size=5, stride=2, padding=0
   - CONV3 is a 2-d convolutional layer with input-channels=20, output-channels=40, filter-size=5, stride=2, padding=0
   - MAXPOOL is a 2-d max pooling layer with filter-size 2, stride 2, padding=0.

   Given an input image of size $3 \times 256 \times 256$, find the size of the output of this CNN. No credit will be given without showing the steps to the solution.

Corner Detection

9. (4 points) Given a Harris quadratic (sum of squared differences) function $E(u, v) = 1000u^2 + 1000v^2 + 10uv$, derive whether a corner, edge, or flat region was detected. For this question, use the following rule for determining whether a corner, edge, or flat region was detected:

   - Corner: both eigenvalues $\lambda_1, \lambda_2 > 100$, where λ_1, λ_2 are the eigenvalues of the Harris covariance matrix (structure matrix)
   - Edge: either $\lambda_1 > 100, \lambda_2 \leq 100$ or $\lambda_1 \leq 100, \lambda_2 > 100$
   - Flat region: both $\lambda_1, \lambda_2 \leq 100$

   No credit will be given without proper justification/derivation.

Homogeneous Coordinates

10. Note that no credit will be given without proper justification.
    (a) (3 points) Prove that, in homogeneous coordinates, a line in a 2D plane can be written as $\hat{l}^T \tilde{x} = 0$, where $\tilde{x} = \lambda(x, y, 1)$ with $\lambda \neq 0$ and $\hat{l} = (a, b, c)$ with $(a, b) \neq (0, 0)$.
    (b) (3 points) Using the previous result, prove that, in homogeneous coordinates, the intersection of two distinct 2D lines with equations $\hat{l}_1^T \tilde{x} = 0$, $\hat{l}_2^T \tilde{x} = 0$ can be obtained by $\hat{x} = \hat{l}_1^T \times \hat{l}_2^T$, where $\hat{l}_1^T \times \hat{l}_2^T$ is the cross-product of $\hat{l}_1^T$ and $\hat{l}_2^T$. For this part, assume that the two lines intersect.
    (c) (2 points) Using the previous result, show that, in homogeneous coordinates, the "intersection" of two parallel lines is at infinity (i.e. a point in homogeneous coordinates with the last coordinate 0).

Interview Brain-teaser

11. (3 points) There are various proven technologies to scan a 3D object in the industry, such as stereo cameras, structured light cameras, and time of flight sensors. In this question, let us see if we can propose a new 3D measurement system based on liquid deformations. If successful, this liquid-based system would be a monocular (single viewpoint) system that does not require ultrafast lasers or circuitry.

    Let's abstract the problem. Imagine you are in an empty room with the following items:

    - A black statue you would like to 3D scan
    - A bucket
    - A few gallons of Milk
    - A measuring cup
    - A ruler
    - An ordinary, fixed-focus camera connected to a computer

    Provide a sketch for how you would obtain the 3D shape of the object. Note that a hint for this question has been provided on Piazza (see post 220).

(Bonus) Perfection is not always desired

12. We wish to train a GAN (i.e. the generator G and discriminator D) using the loss function discussed in the lecture, i.e. $\min_G \max_D \mathbb{E}_{x\sim p_r}[\log(D(x))] + \mathbb{E}_{z\sim p_z}[\log(1 - D(G(z)))]$, where p_r, p_z are the real data and noise distributions, respectively. However, assume that we already have a perfect discriminator D, i.e. $D(\mathbf{x}) = 0$ if $\mathbf{x}$ is a sample generated by the generator, and $D(\mathbf{x}) = 1$ if $\mathbf{x}$ is a sample chosen from the training dataset.

    (a) (2 points) Using this perfect discriminator (which is assumed to be fixed and always perfect), and a randomly initialized generator network as a starting point, will the training process ever result in a usable generator network? Explain using the loss function for the generator.

1. (a) $$KL(p_1||p_2) = \int p_1(x) \log\left(\frac{p_1(x)}{p_2(x)}\right) dx$$

$$= \int p_1(x) \log \frac{\frac{1}{\cancel{\sqrt{2\pi}}\sigma_1} e^{-\frac{1}{2}\left(\frac{x-\mu_1}{\sigma_1}\right)^2}}{\frac{1}{\cancel{\sqrt{2\pi}}\sigma_2} e^{-\frac{1}{2}\left(\frac{x-\mu_2}{\sigma_2}\right)^2}} dx$$

$$= \int p_1(x) \left[-\frac{1}{2}\left(\frac{x-\mu_1}{\sigma_1}\right)^2 + \frac{1}{2}\left(\frac{x-\mu_2}{\sigma_2}\right)^2 - \log \sigma_1 + \log \sigma_2 \right] dx$$

$$= E_1 \left\{ \log\left(\frac{\sigma_2}{\sigma_1}\right) + \frac{1}{2}\left[\left(\frac{x-\mu_2}{\sigma_2}\right)^2 - \left(\frac{x-\mu_1}{\sigma_1}\right)^2 \right] \right\}$$

$$= \log\left(\frac{\sigma_2}{\sigma_1}\right) + \frac{1}{2\sigma_2^2} E_1[(X-\mu_2)^2] - \frac{1}{2\sigma_1^2} E_1[(X-\mu_1)^2]$$

$$= \log\left(\frac{\sigma_2}{\sigma_1}\right) + \frac{\sigma_1^2 + (\mu_1-\mu_2)^2}{2\sigma_2^2} - \frac{1}{2}$$

$$E_1[(X-\mu_1)^2] = E_1[(X-\mu_1+\mu_1-\mu_2)^2]$$

$$= E_1[(X-\mu_1)^2 + 2(X-\mu_1)(\mu_1-\mu_2) + (\mu_1-\mu_2)^2]$$

$$= E_1[(X-\mu_1)^2] + 2(\mu_1-\mu_2)E[X-\mu_1] + (\mu_1-\mu_2)^2$$

$$= \sigma_1^2 + (\mu_1-\mu_2)^2$$

(b) $$KL(p_2||p_1) = \int p_2(x) \log\left(\frac{p_2(x)}{p_1(x)}\right) dx$$

$$= \int p_2(x) \left[\frac{1}{2}\left(\frac{x-\mu_1}{\sigma_1}\right)^2 - \frac{1}{2}\left(\frac{x-\mu_2}{\sigma_2}\right)^2 + \log \sigma_1 - \log \sigma_2 \right] dx$$

$$= E_2 \left\{ \log\left(\frac{\sigma_1}{\sigma_2}\right) + \frac{1}{2}\left[\left(\frac{x-\mu_1}{\sigma_1}\right)^2 - \left(\frac{x-\mu_2}{\sigma_2}\right)^2 \right] \right\}$$

$$= \log\left(\frac{\sigma_1}{\sigma_2}\right) + \frac{1}{2\sigma_1^2} E_2[(X-\mu_1)^2] - \frac{1}{2\sigma_2^2} E_2[(X-\mu_2)^2]$$

$$= \log\frac{\sigma_1}{\sigma_2} + \frac{\sigma_2^2 + (\mu_1-\mu_2)^2}{2\sigma_1^2} - \frac{1}{2}$$

(c) let $f(p_1||p_2) = KL(p_1||p_2) + KL(p_2||p_1)$

then $f(p_2||p_1) = KL(p_2||p_1) + KL(p_1||p_2) = f(p_1||p_2)$

$\Rightarrow$ symmetric w.r.t. p_1, p_2

2. (a) $p(y|x) = \hat{y}^y (1-\hat{y})^{(1-y)}$

$$= \begin{cases} 1-\hat{y} & \text{for } y=0 \\ \hat{y} & \text{for } y=1 \end{cases}$$

equivalently, we could write $p(y|x) = ay+b$

when $y=0$, $p(y|x) = b = 1-\hat{y}$

when $y=1$, $p(y|x) = a+b = \hat{y} \Rightarrow a = 2\hat{y}-1$

thus $p(y|x) = (2\hat{y}-1)y + (1-\hat{y})$

(b) $$\frac{\partial L}{\partial \hat{y}} = \frac{\partial}{\partial \hat{y}}\left((1-2\hat{y})y + (\hat{y}-1)\right)$$

$$= -2y+1$$

(c) No, $L(y,\hat{y}) = -ay-b$ cannot be used to train via SGD.

As seen in (b), $$\frac{\partial L}{\partial \hat{y}} = \begin{cases} 1 & \text{for } y=0 \\ -1 & \text{for } y=1 \end{cases}$$

the gradient does not depend on $\hat{y}$ and thus cannot converge.

3. (a) $\min L(y^*, y) = 0$ minimum possible value is 0

(b) since L is a quadratic function with $\beta > 0$, when $y = y^*$, L has minimum value 0

$$\frac{\partial L}{\partial y} = 2\beta(y - y^*) = 0$$

$$y = y^*$$

(c) $L(y^*; y) = \beta(y-0)^2 = \beta y^2$

$$\frac{\partial L}{\partial y} = 2\beta y$$

(d) $y \to y - \alpha \frac{\partial L}{\partial y} = y - \alpha \cdot 2\beta y = (1 - 2\alpha\beta) y$

(e) during each update, y is multiplied with $1 - 2\alpha\beta$

so after n updates $y = y_0 (1 - 2\alpha\beta)^n$

(f) as long as $0 \leq 1 - 2\alpha\beta < 1$, $(1 - \alpha\beta)^n \to 0$ as $n \to \infty$

$0 \leq 1 - 2\alpha\beta$ $\qquad$ $1 - 2\alpha\beta < 1$

$2\alpha\beta \leq 1$ $\qquad$ $2\alpha\beta > 0$

$\alpha \leq \frac{1}{2\beta}$ $\qquad$ $\alpha > 0$

thus $0 < \alpha \leq \frac{1}{2\beta}$ allows y to converge as $n \to \infty$

4. $\Sigma_F = \begin{pmatrix} 3 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 0 \end{pmatrix} \Rightarrow \sigma_1 = 3, \sigma_2 = 5$

$\Sigma_Q = \mathrm{diag}[\frac{\sigma_1+\sigma_2}{2}, \frac{\sigma_1+\sigma_2}{2}, 0] = \begin{pmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 0 \end{pmatrix}$

thus approximation for essential matrix is:

$$\hat{Q} = U \begin{pmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 0 \end{pmatrix} V^T$$

5. projective transformation include translations, which changes the location of the origin

for instance, say we have a simple projection that only translates the plane $T = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix}$ $t_x, t_y \neq 0$

apply it to origin: $\begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} t_x \\ t_y \\ 1 \end{pmatrix} \Rightarrow \begin{pmatrix} t_x \\ t_y \end{pmatrix}$ in heterogeneous coords

homogeneous coordinate of origin

it is clear that now origin maps to (t_x, t_y)

6. (a) Since people and objects only move sparsely across the scene, we can assume that for the majority of time of each pixel, we capture the static background. As a result, we could apply a median filter in the temporal dimension for every pixel in spatial dimension to get the static background.

(b) The algorithm in (a) does not work since now the background is rotated across time. Instead, we could first use SIFT to find correspondences and apply homography to find the exact transformation that maps the pixels of one frame to the other. Stitch the frames together to form a panorama and apply median filter for every pixel of the panorama across temporal dimension (like in (a)).

7. (a) first examine a toy example

let $I = \begin{pmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{pmatrix}_{3\times 3}$ $\quad K = \begin{pmatrix} k_4 & k_3 \\ k_2 & k_1 \end{pmatrix}_{2\times 2}$

then $I \circledast K = \begin{pmatrix} k_1a_1 + k_2a_2 + k_3a_4 + k_4a_5 & k_1a_2 + k_2a_3 + k_3a_5 + k_4a_6 \\ k_1a_4 + k_2a_5 + k_3a_7 + k_4a_8 & k_1a_5 + k_2a_6 + k_3a_8 + k_4a_9 \end{pmatrix}_{2\times 2}$

$$\underset{4\times 1}{(I \circledast K)'} = \underset{4\times 9}{K'}\underset{9\times 1}{I'} = \begin{pmatrix} k_1 & k_2 & 0 & k_3 & k_4 & 0 & 0 & 0 & 0 \\ 0 & k_1 & k_2 & 0 & k_3 & k_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & k_1 & k_2 & 0 & k_3 & k_4 & 0 \\ 0 & 0 & 0 & 0 & k_1 & k_2 & 0 & k_3 & k_4 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \\ a_9 \end{pmatrix}$$

we could easily extend this to other sizes

also, since convolution is a linear operator, and for each entry of the matrix $I \circledast K$, it is a sum of product between elements in I and filter K. Thus if we flatten $I \circledast K$ and I, it is always possible to find a matrix K' such that $(I \circledast K)' = K'I'$, as long as we fill the entries of K' with proper 0's and values in K.

(b) from (a) we have $(I \circledast K)' = K'I'$, thus

$$K'^T (I \circledast K)' = (K'^T K') I'$$

assume $(K'^T K')^{-1}$ exists

$$(K'^T K')^{-1} K'^T (I \circledast K)' = I'$$

first flatten the $r \times c$ input, then multiply it with the matrix $(K'^T K')^{-1} K'^T$, reshape the result into $h \times w$ as output

(c) from (b), $K'_{inv} = (K'^T K')^{-1} K'^T$

this is under the assumption that $(K'^T K')^{-1}$ exists

however, if $K'^T K'$ is not invertible, K'_{inv} does not exist

8. conv1: height = width = $\frac{256-5}{2}+1 = 126$

$(3, 256, 256) \rightarrow (10, 126, 126)$

maxpool: $(10, 126, 126) \rightarrow (10, 63, 63)$ shrink by 1/2

conv2: height = width = $\frac{63-5}{2}+1 = 30$

$(10, 63, 63) \rightarrow (20, 30, 30)$

maxpool: $(20, 30, 30) \rightarrow (20, 15, 15)$

conv3: height = width = $\frac{15-5}{2}+1 = 6$

$(20, 15, 15) \rightarrow (40, 6, 6)$

the output size is $40 \times 6 \times 6$

9. $E(u,v) = (u \ \ v)\begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix} = m_{11}u^2 + (m_{12}+m_{21})uv + m_{22}v^2$

$= 1000u^2 + 10uv + 1000v^2$

$m_{11} = 1000, \ m_{22} = 1000$

$m_{12} = m_{21} = 5$ $\Rightarrow$ $M = \begin{pmatrix} 1000 & 5 \\ 5 & 1000 \end{pmatrix}$

find eigenvalues of M: $\det\begin{pmatrix} 1000-\lambda & 5 \\ 5 & 1000-\lambda \end{pmatrix} = 0$

$(1000-\lambda)^2 = 25$

$\lambda - 1000 = \pm 5$

$\lambda_1 = 995, \ \lambda_2 = 1005$

since $\lambda_1, \lambda_2 > 100$, this is a corner

10. (a) let $\hat{l}^T\tilde{x}=0$. $\hat{l}=(a,b,c)$, $\tilde{x}=\lambda(x,y,1)$

$$\lambda(ax+by+c)=0 \qquad \lambda, a, b \neq 0$$
$$ax+by+c=0$$
$$by=-ax-c$$
$$y=-\frac{a}{b}x-\frac{c}{b}$$ which is a line in 2D plane $(b\neq 0)$

conversely, a line in 2D plane can also be arranged in the form of $\hat{l}^T\tilde{x}=0$

(b) let $\hat{l}_1=(a,b,c)$ $\hat{l}_2=(d,e,f)$

$$\hat{l}_1\times\hat{l}_2=\begin{pmatrix}\hat{i} & \hat{j} & \hat{k}\\ a & b & c\\ d & e & f\end{pmatrix}=(bf-ce,\ cd-af,\ ae-bd)$$

$\Rightarrow$ $x=\dfrac{bf-ce}{ae-bd}$, $y=\dfrac{cd-af}{ae-bd}$ in 2D coords

$$\hat{l}_1^T\tilde{x}=0 \qquad \hat{l}_2^T\tilde{x}=0$$
$$ax+by+c=0 \qquad dx+ey+f=0$$
$$y=-\frac{a}{b}x-\frac{c}{b} \qquad y=-\frac{d}{e}x-\frac{f}{e}$$

intersection: $\frac{a}{b}x+\frac{c}{b}=\frac{d}{e}x+\frac{f}{e}$
$$aex+ce=bdx+bf$$
$$(bd-ae)x=ce-bf$$
$$x=\frac{ce-bf}{bd-ae}=\frac{bf-ce}{ae-bd}$$

$$ax=-by-c \qquad \frac{a}{d}=\frac{by+c}{ey+f}$$
$$dx=-ey-f$$
$$aey+af=bdy+cd$$
$$y=\frac{cd-af}{ae-bd}$$

this yields the same result as the cross product
thus intersection is also given by $\hat{x} = \hat{l}_1^T \times \hat{l}_2^T$

(c) for parallel lines, directional vectors $\hat{l}_1$ and $\hat{l}_2$ have the same direction

let $\hat{l}_1 = (a, b, 1)$, then $\hat{l}_2 = (a, b, c)$ ← scaling factor

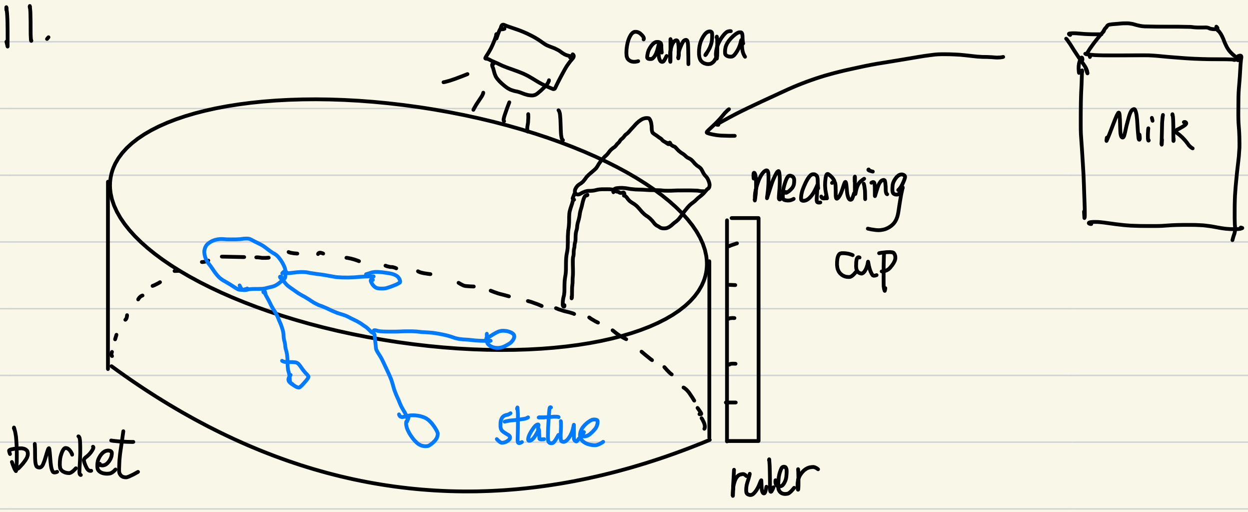
$$\hat{x} = \hat{l}_1 \times \hat{l}_2 = \begin{pmatrix} \hat{i} & \hat{j} & \hat{k} \\ a & b & 1 \\ a & b & c \end{pmatrix} = (bc-b,\ a-ac,\ \underbrace{\cancel{ab-ab}}_{0})$$

when we convert $\hat{x}$ back to heterogeneous coords,

$$x = \frac{bc-b}{0} \to \infty, \quad y = \frac{a-ac}{0} \to \infty$$

so the "intersection" is at infinity

11.



put the statue in the bucket, pour milk slowly with a measuring cup, and take pictures constantly while recording the height of milk with a ruler since milk is an opaquely white liquid, it covers the part of statue beneath the milk level and what camera captures is all the parts above milk

Compare the photos we can see which parts just disappear as we add in a bit of milk, and the height information allows us to locate the depth of these features of the statue, so it's like slicing the statue into many segments in 3D printing

12. Since now we have perfect discriminator D, the optimization problem becomes: $\min_G E_{x\sim P_r}[\log(D(x))] + E_{z\sim P_z}[\log(1-D(G(z)))]$

$$= \min_G P_r(x)\log D(x) + P_z(z)\log(1-D(G(z)))$$

$$= \min_G P_r(x)\cancel{\log 1} + P_z(z)\log\cancel{(1-0)} = 0$$

as a result, $V(D,G)$ is always 0 with D = perfect discriminator, which gives no room for the generator to improve (loss = 0, gradient = 0), this leads to vanishing gradients

no matter how hard the generator tries, the discriminator always finds the fake image, so the training won't result in a usable network