# CS131 Computer Vision: Foundations and Applications
# Practice Final (Solution)

Stanford University
December 11, 2017

Name: _____

SUNet ID: _____@stanford.edu

| Question | Total | **Points** |
|---|---|---|
| Multiple Choice | 11 | |
| True/False | 11 | |
| Filters, features, etc. | 12 | |
| Segmentation and seam carving | 15 | |
| Classification and detection | 10 | |
| Optical flow and tracking | 7 | |
| Total | 66 | |

**Instructions:**

1. This examination contains 15 pages, including this page. You have **three (3) hours** to complete the examination. As a courtesy to your classmates, we ask that you not leave during the last fifteen minutes.
2. For **Multiple-Choice** and **True False** sections, correct answers will get the points listed next to the question. You will receive 0 for all unanswered questions and $-1$ for all incorrect answer. You can choose to not answer questions that you are unsure of to avoid receiving a negative point.
3. For **Short Answers**, you will NOT get negative points for incorrect answers. Partial credit will be assigned for showing your work and reasoning.
4. You may use **one** double-sided 8.5" $\times$ 11" **hand-written** page with notes that you have prepared. You may not use any other resources, including lecture notes, books, other students or other engineers. These notes must be submitted along with your booklet.
5. You may use a calculator however, you should not require one. You may not share a calculator with anyone.
6. Please sign the below Honor Code statement.

In recognition of and in the spirit of the Stanford University Honor Code, I certify that I will neither give nor receive unpermitted aid on this examination.

Signature: _____

## Multiple choice answers (11points)

1. **RANSAC vs Least squares (1 point).** What are the benefits of RANSAC compared to the least squares method? *Circle all that apply.*
    A. RANSAC is faster to compute in all cases
    B. RANSAC has a closed form solution
    C. RANSAC is more robust to outliers
    D. RANSAC handles measurements with small Gaussian noise better

2. **SIFT (1 point).** What does each entry in a 128-dimensional SIFT feature vector represent? (Choose the best answer)
    A. A principal component.
    B. A sum of pixel values over a small image patch.
    C. One bin in a histogram of gradient directions.
    D. The number of pixels whose gradient magnitude falls within some range.

3. **Clustering (1 point).** Which of the following statement is true for $k$-means and HAC clustering algorithm? (Choose the best answer)
    A. $k$-means and HAC are both sensitive to cluster center initialization.
    B. $k$-means and HAC can lead to different clusters when applied to the same data.
    C. $k$-means works well for non-spherical clusters.
    D. HAC starts with all examples in the same cluster, then each cluster is split until we have the desired number of predetermined clusters.

4. **Video Tracking (1 point).** What kinds of image regions are hard to track? *(Circle all that apply)*:
    A. low texture regions like sky
    B. Edges of objects
    C. Corners of objects
    D. Rotating spheres

5. **Optical flow (1 point).**What are the methods for estimating optical flow that we covered in class? *Circle all that apply.*
    A. Seam carving.
    B. Gibbs random field motion estimation.
    C. Optical flow equation and Second order derivatives of optical flow field.
    D. Hidden Markov Model.

6. **Hessian (1 point).** Given a vector in $\mathcal{R}^n$ (implying that the vector is $n$-dimensional), its Hessian is in:
    A. undefined
    B. $\mathcal{R}^n$
    C. $\mathcal{R}^{n \times n}$
    D. $\mathcal{R}^{n^2}$

7. **Similarity transformation (2 points).** Which of the following always hold(s) under an similarity transformation? *Circle all that apply.*
    A. Parallel lines will remain parallel.

B. The ratio between the two areas or two polygons will remain the same.

C. Perpendicular lines will remain perpendicular.

D. The angle between two line segments will remain the same.

8. **Scale invariance (2 points).** Which of the following representations of an image region are scale invariant? *Circle all that apply.*

   A. Bag of words model with SIFT features

   B. Spatial Pyramid Model with SIFT features

   C. A HOG template model

   D. Deformable Parts Model

9. **Optical flow (1 point).** Optical flow is problematic in which conditions? *Circle all that apply.*

   A. In homogeneous image areas.

   B. In textured image areas.

   C. At image edges.

   D. At the boundaries of moving objects.

## True and False (11points)

1. ____ (1 point.) A smoothing filter sums to 1.
2. ____ (1 point.) A scaling matrix by $a$, $b$, $c$ in the $x$, $y$, and $z$ directions, respectively, has the transformation matrix: $T = \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & \frac{1}{abc} \end{bmatrix}$.
3. ____ (1 point.) In figure 1, we applied a Gaussian filter to the original image and computed the derivative in $x$ and $y$ directions. **True or False:** Result 1 is the $x$-derivative of Gaussian, and result 2 is the $y$-derivative of Gaussian.

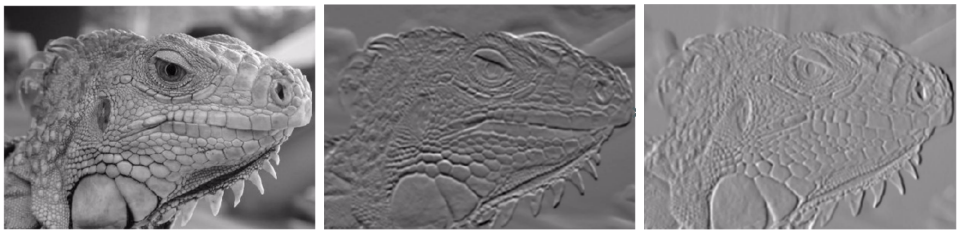


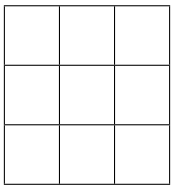
Figure 1: **Left**: Original image; **Center**: Result 1; **Right**: Result 2

4. ____ (1 point.) If you compute optical flow in a video sequence, a nonzero optical flow vector always indicates real movement of an object in the scene depicted by the video.
5. ____ (1 point.) A set of eigenfaces can be generated by performing principal component analysis (PCA) on a large set of images depicting different human faces. Any human face from this large set can be considered to be a combination of these standard faces.
6. ____ (1 point.) Pyramidal Lucas-Kanade method can track large motions between frames while Lucas-Kanade without the pyramids can not.
7. ____ (1 point.) When using Harris detector, an image region is considered to be a corner if both $\lambda_1$ and $\lambda_2$ are small. ($\lambda_1$ and $\lambda_2$ are the eigenvalues of the second moment matrix.)
8. ____ (1 point.) Even without using non-maximum suppression, we know that all pixels with gradient magnitudes above the 'high' threshold will be considered an edgeb by a Canny edge detector.
9. ____ (1 point.) Optical illusions are NOT a type of optical flow.
10. ____ (1 point.) In texture free regions, there is no optical flow.
11. ____ (1 point.) Motion could be used to improve video resolution quality.

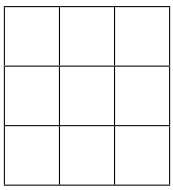
## Short Answers 1: Filters, edges, corners, keypoints and descriptors (12points.)

1. **Filters (4 points).** Give ANY 3x3 example of each of the following types of image convolution filters:

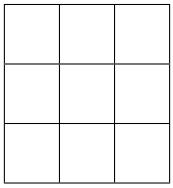
   a. **(1 point.)** A dimming filter (only decrease the image intensity)

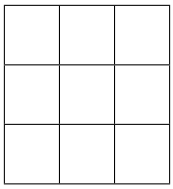
   b. **(1 point.)** Approximation for difference of Gaussian filter

   c. **(1 point.)** A filter for detecting diagonal edge from top-right to bottom-left

   d. **(1 point.)** A filter that shifts pixels to the right by one pixel

2. **Median filter (4 points).** A "Median Filter" operates over a window by selecting the median intensity in the window.

    1. **(1 point).** What advantage does a median filter have over a mean filter?

    2. **(3 points).** Prove that the median filter can be written as a convolution operation or explain with an example why it can not.

3. **Cross-correlation (4 points)**. Prove that the superposition principle holds for cross-correlation or provide a counterexample.

   Recall that the cross-correlation between $f$ and a template $h$ is function $S[f] = f \star h$ defined by:

$$(S[f])[m,n] = (f \star h)[m,n] \stackrel{\text{def}}{=} \sum_k \sum_l f[m+k, n+l]h[k,l]$$

   .

   The superposition principle for system $S$ is:

$$S[\sum_i \alpha_i f_i] = \sum_i \alpha_i S[f_i]$$

   . where $S$ is the the cross correlation system operator associated with filter $h$:

$$S[f] = f \star h$$

   The $\alpha_i$ are all constants in $\mathbb{R}$, and the $f_i$ are images defined on $\mathbb{R}^2$.

## Short Answers 2: Segmentation and seam carving (15points.)

1. **$k$-means (4 points.)** A distance function $d$ on $\mathbb{R}^n$ is said to be *invariant* under a transformation $\phi : \mathbb{R}^n \to \mathbb{R}^n$ if

$$\forall x, y \in \mathbb{R}^n, \quad d(x, y) = d(\phi(x), \phi(y))$$

A cluster center function $\mu$ is *invariant* to a transformation $\phi : \mathbb{R}^n \to \mathbb{R}^n$ if

$$\phi(\mu(x^{(1)}, \ldots, x^{(m)})) = \mu(\phi(x^{(1)}), \ldots, \phi(x^{(m)}))$$

for all sets of points $x^{(1)}, \ldots, x^{(m)} \in \mathbb{R}^n$.

A clustering algorithm is said to be invariant to a transformation $\phi$ if its choice of clusters is not affected when all points $x$ are transformed to $\phi(x)$

**Prove or reject with a counter-example**: $k$-means algorithm is invariant to a transformation $\phi$ if and only if the distance and cluster center function are invariant to $\phi$.

2. **Seam Carving (11 point).** You have implemented seam carving for image resizing in one direction (vertical or horizontal). Let $I$ be an $n \times m$ image and define a vertical seam to be:

$$\mathbf{s^y} = \{(i, y(i)\}_{i=1}^{n}, s.t. \forall i, |y(i) - y(i-1)| \leq 1,$$

where $y$ is a mapping $y : [1, ..., n] \rightarrow [1, ..., m]$. Let $e(i, j)$ define the energy of the pixel at location $(i, j)$ of image $I$. Now, we can define the cost of a seam as $Cost(\mathbf{s^y}) = \sum_{i=1}^{n} e(i, y(i))$. We look for the optimal (vertical) seam $\mathbf{s^{y*}}$ that minimizes this seam cost:

$$\mathbf{s^{y*}} = \min_{\mathbf{s^y}} Cost(\mathbf{s^y})$$

(a) **(2 points).** What is the time complexity for brute-force searching strategy to find the optimal seam? (hint: think about how many possible vertical seams are there in $n \times m$ image.)

(b) **(4 points).** The optimal seam can be found efficiently using dynamic programming. The algorithm traverses the image from the first row to the last row, and at each row $i$, it computes the cumulative minimum cost $M(i, j)$ for $\forall j$. Recall that $M(i, j)$ is the minimum cost of a seam from the top row to the pixel at $(i, j)$. Define the recurrence relationship that seam carving uses to calculate $M(i, j)$. In other words, define $M(i, j)$ using the functions $M(\cdot)$ and $e(\cdot)$.

(c) **(5 points).** Suppose we want to resize an image $I$ of size $n \times m$ to size $n' \times m'$, where $n' < n$ and $m' < m$. We can accomplish this by first removing vertical seams and then removing horizontal seams. Or we first remove the horizontal seams followed by the vertical seams. This begs the question of what is the correct order of removing seams. Remove vertical or horizontal seams first? Or alternate between the two? Here, we will derive the optimal list of seams to remove by finding the best vertical or horizontal seam to remove at every step.

From the previous parts of the question, we have defined $\mathbf{s^y}$ to be a vertical seam. Since, we will need to remove multiple vertical seams, let's define a new variable $\mathbf{s_t^y}$, which is the vertical seam we want to remove at step $t$. Similarly, let's define that $\mathbf{s_t^x}$ as the horizontal seam we want to remove step $t$. At every step of this problem, we need to choose whether to remove the optimal vertical seam or the optimal horizontal seam. Let $\alpha_t$ be the variable that determines which seam to remove. When $\alpha_t = 0$, we will remove the vertical seam at step $t$ and when $\alpha_t = 1$, we will remove the horizontal seam. We can write this as the following objective function:

$$\min_{\mathbf{s_t^x}, \mathbf{s_t^y}, \alpha_\mathbf{t}} \sum_{t=1}^{(n-n')+(m-m')} Cost(\alpha_\mathbf{t} \mathbf{s_t^x} + (1 - \alpha_\mathbf{t}) \mathbf{s_t^y})$$

$$\text{such that} \sum_{t=1}^{(n-n')+(m-m')} \alpha_\mathbf{t} = n - n'$$

$$\sum_{t=1}^{(n-n')+(m-m')} (1 - \alpha_\mathbf{t}) = m - m'$$

Given that you already know how to find $\mathbf{s_t^x}$ and $\mathbf{s_t^y}$ at every step, describe how you would use dynamic programming to find the values $\alpha_\mathbf{t}$. Let $\mathbf{T}(i, j)$ be the optimal cost of removing $i$ rows and $j$ columns from the image $I$. Write a recurrence relationship for $\mathbf{T}(i, j)$ using the functions $\mathbf{T}(\cdot)$ and $Cost(\cdot)$.

## Short Answers 3: Classification and detection (10points.)

1. **Eigenpenguins (4 points).**

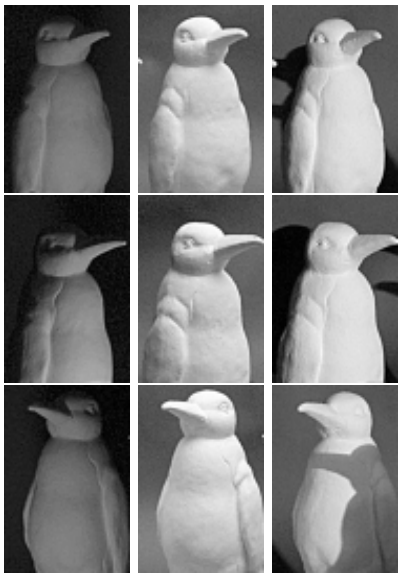


Figure 2: A database of penguins

You have collected a small dataset of penguin statues from different angles and under different lighting conditions, shown in Figure 2. In this figure note that each row contains images of penguin statues taken from the same angle, and each column contains penguin statues under the same lighting conditions.

You would like to use this dataset to recognize the angle from which a photo of a penguin statue has been taken, and you decide to use the Eigenfaces (Eigenpenguins?) algorithm. More concretely, you want to determine the angle from which the image in the upper left corner was taken, using the other images as a training dataset. You use the other images to choose a low-dimensional Eigenpenguin space, project all of the images into this space, and declare that the angle of the query image is the angle of its nearest neighbor in Eigenpenguin space.

Do you expect this algorithm to work correctly for this dataset? Justify your answer or explain why it isn't a good algorithm choice.

2. **Image compression (6 points).**

You have a big database of images of dogs, with $100,000$ images, each of size $(200, 200, 3)$. The raw pixel values are stored on your database (you don't have jpeg), which means that you currently store $100,000 \times 200 \times 200 \times 3$ floats in memory. You want to to reduce the cost of storing all the data.

(a) **(2 points).** Describe the algorithm you will use to compress / decompress images.

(b) **(2 points).** How will you decide how much you can compress? Let's assume that you you want to maintain $x\%$ variance of your data.

(c) **(1 point).** What will the final size of the compressed database be?

(d) **(1 point).** What kind of additional information will you have to store to allow you to compress and decompress images? What is the size of that additional information?

3. **kNN Boundaries (4 points).** Figure 3 shows a training set of 14 examples, with 5 example for class 1 (with the crosses) and 3 examples for class 2 (with the circles).

   If we apply k-nearest neighbors, we can classify any testing point into class 1 or class 2. If we do that for every point in $\mathbb{R}^2$, we obtain decision boundaries which delimit zones of $\mathbb{R}^2$ where the classification is either 1 or 2.

   (a) **((2 points).** Draw on the image (figure 3) the decision boundaries for $k = 1$. Use **L2 distance** to get nearest neighbors.

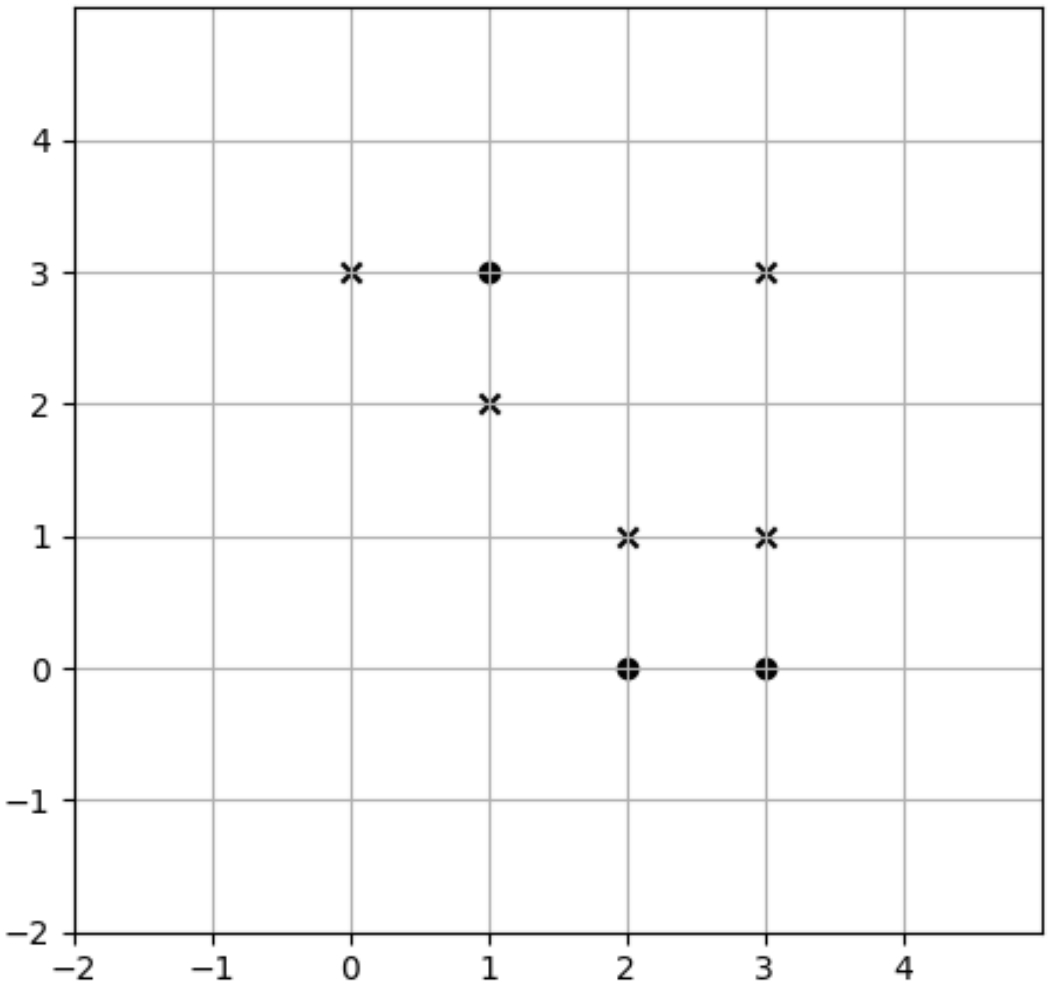



Figure 3: Decision boundaries for $k$-NN with $k = 1$ and $L_2$ distance

(b) **((2 points).** Draw on the image (figure 4) the decision boundaries for $k = 3$. Use **L2 distance** to get nearest neighbors.

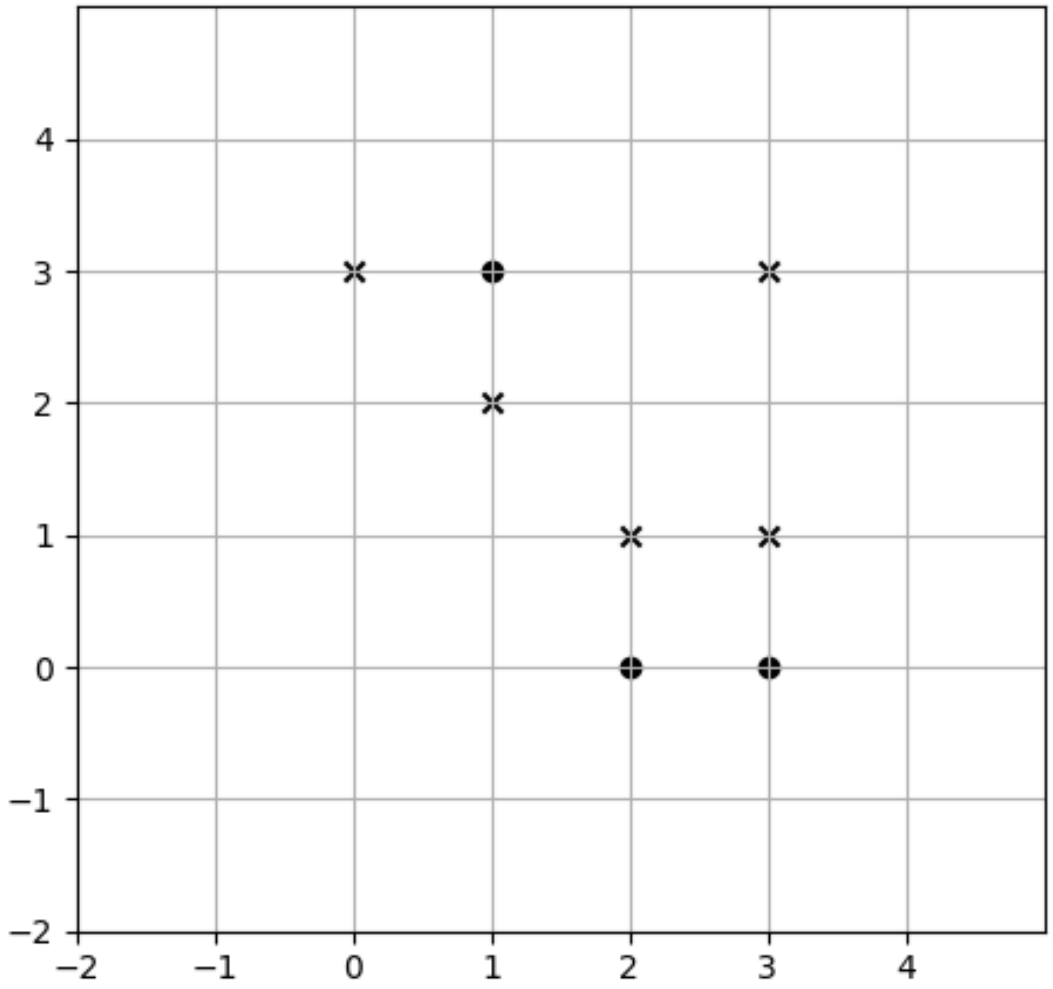



Figure 4: Decision boundaries for $k$-NN with $k = 3$ and $L_2$ distance

## Short Answers 4: Optical flow and tracking (7points).

1. **KLT tracker (7 questions).** To update the change in parameters in KLT tracker, we derived the following equation:

$$\Delta p = H^{-1} \sum_{\mathbf{x}} \left[ \nabla I \frac{\partial W}{\partial p} \right]^T [T(x) - I(W(\mathbf{x}, \mathbf{p_0}))]$$

where

$$H = \sum_{\mathbf{x}} \left[ \nabla I \frac{\partial W}{\partial p} \right]^T \left[ \nabla I \frac{\partial W}{\partial p} \right]$$

.

Given that the 2D motion we are trying to track is a similarity motion parameterized by

$$p = \begin{bmatrix} a \\ b_1 \\ b_2 \end{bmatrix},$$

such that $x' = ax + b_1$ and $y' = ay + b_2$, solve the following two parts:

(a) **(3 points).** Derive the Jacobian $\frac{\partial W}{\partial p}$.

(b) **(4 points).** Derive the $H$ matrix in terms of image derivatives ($I_x$ and $I_y$) and the pixel locations ($x$ and $y$).