Department of Electrical & Computer Engineering, NYU Tandon School of Engineering — Spring 2018
EL6123 --- IMAGE AND VIDEO PROCESSING — Yao Wang

Exam

Closed book, 2 sheet of notes (double sided) allowed. No peeking into neighbors or unauthorized notes. No calculator or any electronics devices allowed. Cheating will result in getting an F on the course. Make sure you write your name and ID on the cover of the blue book. Write your answer in the blue book (or on the problem sheet when space is provided on the problem sheet).

Your Name ________________________________ **ID** ____________________________

1. (10pt) The histograms of three images are illustrated in Figs. (a) to (c). Choose one of the three transformations given in Figs. (d) to (f) such that the transformed image has a nearly flat histogram.

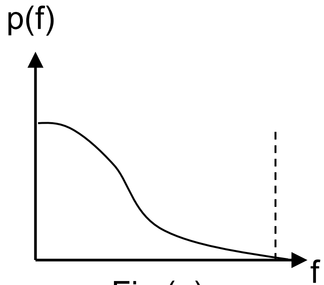


Fig.(a)

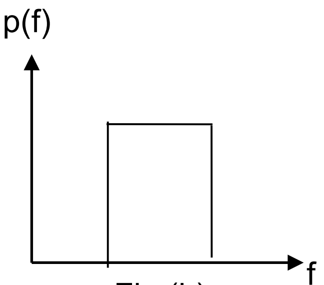


Fig.(b)

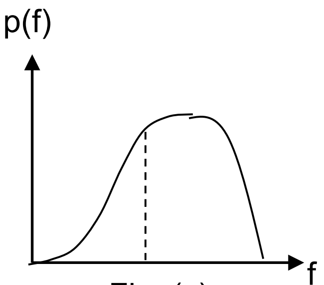


Fig. (c)

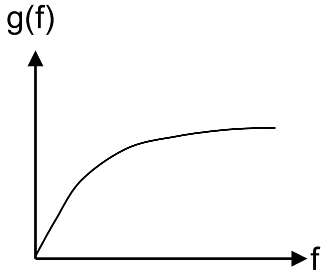


Fig.(d)

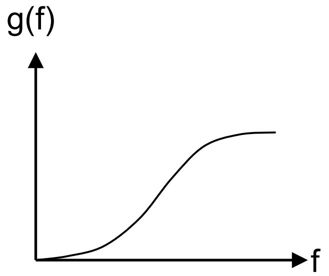


Fig.(e)

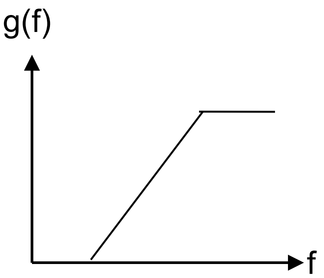


Fig.(f)

Answer:
Fig a → Fig d (4pt)
Fig b → Fig f (2pt)
Fig c → Fig e (4pt)

2. (10 pt) A 2D filter H is given below, where the center position corresponds to m=n=0. (a, 6pt) The filter H can be decomposed into the sum of two filters as shown above. Is each filter (*H, H_1 and H_2*) separable? If so, give the one dimensional filters in horizontal (*y*) and vertical (*x*) directions. Also, based on the filter coefficients, can you tell what is the function of each filter (*H, H_1 and H_2*). (4) Determine the DSFT *H(u,v)* of the filter H by computing the DSFT of the subfilters *H_1 and H_2*.

$$H = \frac{1}{8}\begin{bmatrix} -1 & 2 & -1 \\ 2 & 4 & 2 \\ -1 & 2 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \frac{1}{8}\begin{bmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{bmatrix} = H_1 + H_2$$

Answer:

H is not separable(1pt). H_1 and H_2 are separable (2pt each with the formula).

$$H_1 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$H_2 = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & -2 & 1 \end{bmatrix} \times \frac{1}{\sqrt{8}} \begin{bmatrix} -1 \\ 2 \\ -1 \end{bmatrix}$$

H_1 is an all-pass filter and keeps the images as is. H_2 is a high-pass filter in both directions and can detect edges. H is high emphasis filter and is used for sharpening the image (1pt each).

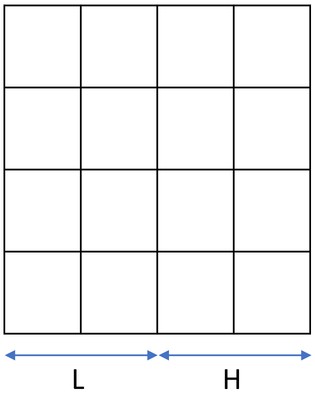
$h_{1x} = h_{1y} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \xrightarrow{yields} H_1(u) = H_1(v) = 1 \xrightarrow{yields} H_1(u,v) = 1$ (1pt)

$h_{2x} = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & -2 & 1 \end{bmatrix} \xrightarrow{yields} H_2(u) = \frac{1}{\sqrt{8}} (2\cos(2\pi u) - 2)$ and $h_{2y} = \frac{1}{\sqrt{8}} \begin{bmatrix} -1 & 2 & -1 \end{bmatrix} \xrightarrow{yields} H_2(v) = \frac{1}{\sqrt{8}} (2\cos(2\pi v) - 2)$ (1pt)

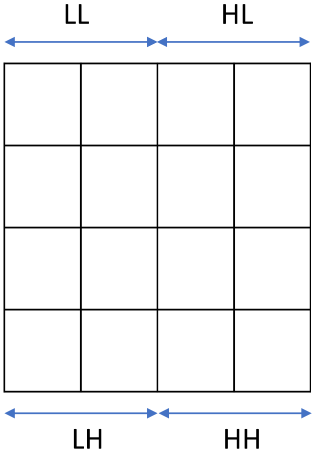
As a result we have, $H_2(u,v) = H_2(u) \times H_2(v)$ and $H(u,v) = 1 + \frac{1}{\sqrt{8}} (2\cos(2\pi u) - 2) \times \frac{1}{\sqrt{8}} (2\cos(2\pi v) - 2)$ (1pt)

3. (10pt) For the image given below, determine its 1-level wavelet decomposition using Haar wavelets. You should use separable processing. First generate two subbands corresponding to row wise decomposition (generating images L and H), and then apply wavelet decomposition column wise to generate 4 subimages (LL, LH, HL, HH). For simplicity, you can assume the analysis stage simply uses a filter of [1 1] and [-1, 1]. That is, given two samples A and B, the low band signal is A+B, the high band signal is A-B. Draw the resulting images in the figures given below.

1	1	1	1
1	1	2	2
1	2	1	2
1	2	2	1



Row wise decomposition



Column wise decomposition

2	2	0	0
2	4	0	0
3	3	1	1
3	3	1	-1

4	6	0	0
6	6	2	0
0	2	0	0
0	0	0	-2

Left image is after row-wise decomposition, right image is after column wise decomposition.
Following was also accepted which would be the result if you didn't flip the filters

2	2	0	0
2	4	0	0
3	3	-1	-1
3	3	-1	1

4	6	0	0
6	6	-2	0
0	-2	0	0
0	0	0	-2

Normalization by $\frac{1}{\sqrt{2}}$ was also accepted. (2.5pt each operation)

4. (10 pt) Magnetic Resonance Imaging (MRI) works by (with some approximations) measuring the 2D Discrete Fourier Transform of each slice of an organ. If all samples in the Fourier space (known as k-space) are measured, the image slice can be easily reconstructed by using Inverse DFT. However, imaging the complete k-space takes long time. To reduce the scan time, it is desirable to measure only a subset of points in the k-space (the total number of samples is significantly less than the total number of pixels). Your task is to recover the image slice $\boldsymbol{x}$ from the measured subset of DTFT coefficients, $\boldsymbol{y}$. Obviously, this is an underdetermined problem. One prior knowledge you can use is that the absolute difference between horizontally adjacent pixels and that between two vertically adjacent pixels are typically small (large only when there are strong edges). Describe how would you formulate this as a quadratic optimization problem and derive a closed form solution. Please define all the notations you use.
Hint: You could think of $\boldsymbol{x}$ as a 1D vector reshaped from a 2D image row by row, and similarly $\boldsymbol{y}$ a 1D vector consisting of the measured DFT coefficients. You can write that $\boldsymbol{y} = \boldsymbol{Fx}$, but you have to stay clearly how to define the matrix $\boldsymbol{F}$ (you just need to say what is the physical meaning of each row of $\boldsymbol{F}$). Also, a vector consisting of the differences between every two horizontally adjacent pixels can be written as $\boldsymbol{Hx}$. Similarly, the vector corresponding to vertical differences can be written as $\boldsymbol{Vx}$. You should define clearly matrices $\boldsymbol{H}$ and $\boldsymbol{V}$.

Solution: The i-th row of matrix F would include elements that corresponds to the Fourier transform basis for the i-th DFT coefficient captured. More specifically (not required), because the 2D FFT is defined as $F(k,l) = \frac{1}{\sqrt{M\,N}}\sum_{m,n} x(m,n)\exp\left\{2\pi\left(\frac{mk}{M}+\frac{nl}{N}\right)\right\}$, if the i-th row corresponds the coefficient index (k,l), then the column that corresponds to pixel (m,n) would have the complex entry of $\frac{1}{\sqrt{M\,N}}\exp\left\{2\pi\left(\frac{mk}{M}+\frac{nl}{N}\right)\right\}$.
If we order the image in a row by row order to a 1D vector, then the horizontal difference between two adjacent pixels in the same row will correspond to a weighted average of two adjacent pixels, one with weight 1 and another with weight 0. Similarly, the vertical different between two vertically aligned pixels in two adjacent rows will correspond to one pixel with weight 1, and another pixel that is N pixels away (N= image width) with weight -1. The matrix H and V are illustrated in the figure below.

To solve for x while making use of the fact that each component Hx, and Vx should have small magnitude while satisfying the data constraint that Fx=y. This can be achieved by minimizing an objective function with the form

$$J(x) = \|Fx - y\|^2 + \lambda(\|Hx\|^2 + \|Vx\|^2)$$

This function is a quadratic function of x, and the minimum can be found by setting the derivative to zero, yielding:

$$F^T(Fx - y) + \lambda\,(H^T Hx + V^T Vx) = 0 \rightarrow x = \left(F^T F + \lambda\,(H^T H + V^T V)\right)^{-1}(F^T y)$$

In practice one has to choose the parameter λ properly. Also, instead of using matrix inversion, other more efficient algorithms may be used to find the solution. (2 pt for F matrix, 3 pt for H and V, 3 pt for the optimization problem formulation (L(x)), 2 pt for the solution).

Instead of using the l2 norm on Hx and Vx, a better solution is to use the l1 norm on Hx and Vx, changing the objective function to

$$J(x) = \|Fx - y\|^2 + \lambda(\|Hx\|_1 \quad + \|Vx\|_1 \ )$$

This problem is convex but not quadratic. It is much harder to solve. But one can use an iterative algorithm to solve it. This is not required. Students who gave the above as the objective function and mention the correct approach for solving such a problem will get full credits.

The solution above does not guarantee that the data constraint is met exactly, i.e. Fx=y. To make that happen, we should solve the following constrained optimization problem: (Not required)

$$\text{Minimize } \|Hx\|^2 + \|Vx\|^2$$
$$\text{Subject to } Fx = y$$

This problem can be solved using the Lagrange multiplier method. (Not required)

$$\text{Minimize } L(x) = \|Hx\|^2 + \|Vx\|^2 + \lambda^T(Fx - y)$$
$$\text{Subject to } Fx = y \quad (1)$$

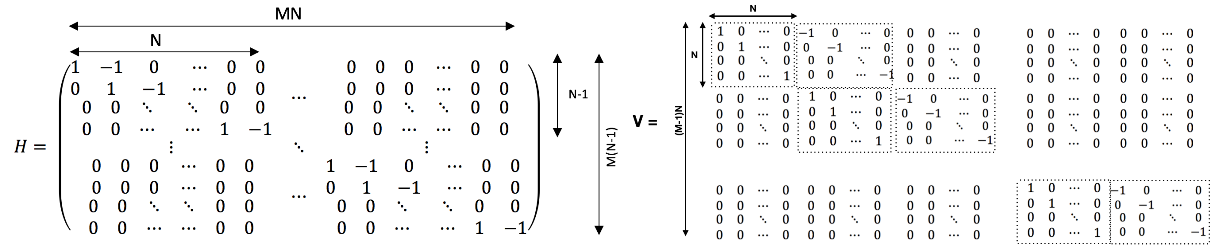
Setting derivative of L(x) with respect to x yields

$$H^T Hx + V^T Vx + F^T\lambda \ = 0 \ (2)$$

Eqs (1) and (2) can be combined as a linear equation of an extended variable containing both x and λ:

$$\begin{bmatrix} H^T H + V^T V & F^T \\ F & 0 \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} y \\ 0 \end{bmatrix}$$

Students formulating the problem as above correctly will get full credits. (Note that some of you has solution that requires the inverse of $(H^T H + V^T V)$. This matrix may not be invertible. But you are given full credit)



Note for the horizontal difference, for each row with N pixels, you only need to compute difference at N-1 pixels positions (1 to N-1), to avoid taking difference with pixels outside the right boundary. Therefore, you have M(N-1) rows in total in matrix H. For vertical difference, for each column, you only need to compute difference at N-1 pixels positions (rows 1 to N-1), to avoid taking difference with pixels outside the bottom boundary. Therefore, you have (M-1)N rows in total in matrix V. These details are not required. As long as your matrix is approximately correct, you get the full credit.

5. (10 pt) We would like to use the histogram of oriented gradient (HoG) to describe an image patch. a) (8pt) Given the image patch below, generate its gradient images Ix and Iy and the gradient orientation image Io and finally the HoG. For pixels where the gradient orientation is undefined, use a notation NA. Write the resulting images in the figure below. Assume all possible orientations are quantized to only 8 directions (0, 45, 90, 135, 180, 225, 270, 315). Please use the simple difference operator to determine the image gradient: Ix(m,n)=I(m,n)-I(m,n-1); Iy(m,n)=I(m,n)-I(m-1,n). You can assume pixels outside the patch have values of 0. (b) (2pt) Is the HoG (after normalization) invariant to image contrast (i.e. the dynamic range of the gray values)? Is it invariant to image rotation?

0	1	0	0
0	1	1	0
0	0	1	1
0	0	0	0

Image I

Gradient Ix

Gradient Iy

Gradient Orientation Io

Answer:

0	1	-1	0
0	1	0	-1
0	0	1	0
0	0	0	0

0	1	0	0
0	0	1	0
0	-1	0	1
0	0	-1	-1

NA	45	180	NA
NA	0	90	180
NA	270	0	90
NA	NA	270	270

(2pt,2pt,3pt) Histogram: 0→2, 45→ 1, 90→2, 180→2, 270→3, other → 0 (1pt)

HoG after normalization is invariant to changes in the dynamic range of gray values(1pt) but it is not invariant to rotation (1pt).

6. (10pt) Suppose you would like to segment an image into K regions so that pixels with similar colors are put into the same region. Assuming pixel n is described by a feature vector $\mathbf{f}_n$. a) Describe the K-means algorithm that you could use to perform the segmentation. Describe it in a ``pseudo code''. b) Compared to K-means, what are the advantages of the GMM algorithm? List two advantages.

Solution:

a) (7pt) We will call the feature vectors corresponding to all the pixels as samples. The K-means algorithm follows the following iterative procedure:

```
Initialize: Choose K samples as the initial centroids. Converge=False
While (converge==false)
    1) For each sample, compute its distance (can be in terms of sum of squared errors in each feature element) with each centroid to determine the centroid that it is closest to. If it is closest to centroid k, then this sample is assigned to cluster k. Repeat for all samples. (assignment update: 2pt)
    2) Calculate the mean of all samples in the same cluster. Use the result as the new centroid for this cluster. Repeat for all K clusters. (centroid update: 2pt)
    3) Calculate the mean square error between each original sample and its assigned cluster centroid.
    4) If the new error is very close to the old error, Converge=True.
```

Note: i) If you set a maximum number of iterations and always run this number of iterations, no point will be deducted. Ii) You do not have to write down how to calculate the distance and centroid. Iii) you do not have to say how to choose the initial centroids. But you need to say that you will choose K initial centroids.

b) (3pt) K-means essentially assume the underlying samples follow a K-mixture model, where each mixture has an isotropic distribution (feature elements are i.i.d. and have same variance), and all the mixtures have similar probability. GMM can explicitly consider the situation where the features are not i.i.d. (e.g. ovals vs. circles) and mixtures can have different probabilities. When the actual sample distribution have non-i.i.d. feature elements and some clusters are more popular than the others, GMM based clustering method (EM algorithm) is likely to perform better.
K-means uses hard assignment to clusters whereas GMM has soft assignment based on probability of each sample being assigned to different clusters.)

7. (15pt) You are given two images taken under different view angles, and you want to align them. Assume that there are quite significant variation in the ambient illumination when the two pictures are taken. Furthermore, assume that the underlying scene is quite far and can be approximated by a plane. a) (2pt) What would be good model that you could use to describe the geometric mapping function between the two images? b) (3pt) Would you use a feature-based or intensity-based approach to determine the mapping function? why? c) (10pt) Describe the steps that you would use to find the mapping parameters.

(a) (2pt) The plane homography.

(b) (3pt) One cannot use intensity based approach because there are significant variation in the ambient illumination. Should use a feature-based approach.

(c) i) we need to find some feature points in each image, e.g. using the Harris corner detector or SIFT feature detection. ii) we form a descriptor for each detected feature based on a small neighborhood surrounding the feature points, e.g. using the HOG descriptor. iii) we find an initial set of matching features between the two images based on the descriptors. For example, for each feature in the first image, compare its descriptor with the descriptors of all features in the second image and find the one that is closest. iv) Apply the RANSAC algorithm to remove outliers among the matching pairs resulting from iii). RANSAC will find a largest set of matching pairs that can be related by a homography model and the corresponding model parameters. (see matrix below). v) Using the homography mapping determined in Step iv) to warp one image to another. (There are 5 steps, 2 pts for each step, note that you need to write out the homography transformation parameter matrix in the iv) step.)

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_0 \\ b_1 & b_2 & b_0 \\ c_1 & c_2 & c_0 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \Rightarrow \begin{matrix} a_0 + a_1 u + a_2 v - c_1 u x - c_2 v x = x \\ b_0 + b_1 u + b_2 v - c_1 u y - c_2 v y = y \end{matrix}$$

$$\begin{bmatrix} \dots \\ \dots \\ 1 & u_n & v_n & 0 & 0 & 0 & -u_n x_n & -v_n x_n \\ 0 & 0 & 0 & 1 & u_n & v_n & -u_n y_n & -v_n y_n \\ \dots \\ \dots \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ b_0 \\ b_1 \\ b_2 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} \dots \\ \dots \\ x_n \\ y_n \\ \dots \\ \dots \end{bmatrix} \Rightarrow Aa = x$$

8. (15pt) Consider block wise motion estimation between two frames. a) (5pt) Let us first consider the exhaustive block matching algorithm (EBMA) algorithm. Suppose the image size is WxH. Block size is BxB. You perform integer-pel accuracy search with a search range of R. What is the total number of operations? (For simplicity, assume the number of operations for calculating the error between two blocks is B^2). b)(5pt) Now consider the hierarchical block matching algorithm (HBMA). Suppose you use two resolution level. At the top level, you use integer accuracy search and the search range at the top level is R/2. You get the initial motion vector at the bottom level by interpolating the motion vector obtained at the top level and you use half-pel accuracy search at the bottom level with a search range of 1. What is the total number of operations? You can ignore the computation needed to generate the low resolution images and for interpolation between image levels and for halfpel search. c) (5pt) Between EBMA and HBMA, which method has less computation? Which one is expected to yield better motion estimation?

Solution

a) (5pt) There are W/B x H/B blocks. Each block has to be compared with $(2R+1)^2$ candidates. Each comparison takes B^2 operations. The total number of operations $N1 = WH/B^2\ (2R+1)^2\ B^2 = WH\ (2R+1)^{2.}$

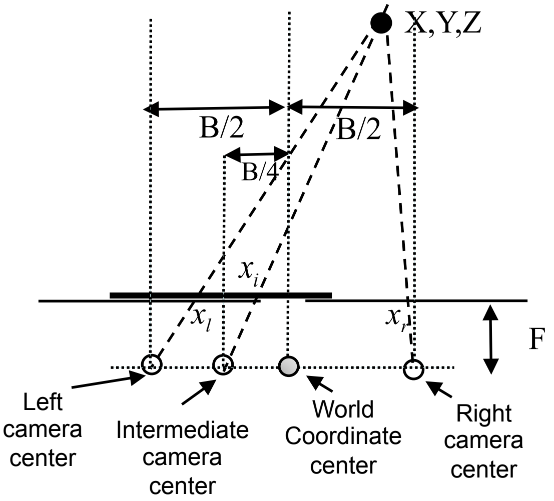
b) (5pt) For the top layer, the number of operations = $W/2\ H/2\ (2\ R/2+1)^2 = W\ H\ (R+1)^2/4$. For the bottom layer, a search range of 1 with half-pel accuracy search requires 5^2 comparisons for each block. So the total operation number = W H 25.
So the total number of operations $N2 = WH\ ((R+1)^2/4 + 25)$

c) (5pt) For R>>1, N1 \approx 4 W H R^2. N2 \approx W H R^2 / 4. Therefore HBMA will take 1/16 of the operation of EBMA with this particular set up. HBMA is likely to yield lower prediction PSNR than EBMA since they have the same overall search range. But HBMA is likely to yield a smoother motion field, which can be more accurate.

9. (10pt) Consider a parallel stereo imaging system with baseline distance *B* and focus length *F* (see below). Suppose that for an object point at world coordinate *(X,Y,Z)*, its image position in the left and right view are (x_l, y) and (x_r, y), respectively.

a. (2 pt) Describe how to estimate the 3D position *X,Y,Z* from x_l, x_r, y.

b. (8 pt) Suppose we want to generate an intermediate view, whose camera center has a distance of *B*/4 away from the world coordinate origin, as shown below. Describe an algorithm that you will use to generate the intermediate view. For this problem, you could ignore the difficulty that the possible position in the image to be generated for a given pixel in the left or right image may not be an integer.



Solution:

a) Using perspective projection model, $x_l = \frac{F\left(X+\frac{B}{2}\right)}{Z}, x_r = \frac{F\left(X-\frac{B}{2}\right)}{Z}$. Therefore, $x_l - x_r = \frac{FB}{Z}$. Therefore, we can recover Z from the disparity using $Z = \frac{FB}{x_l - x_r}$. Then we can recover X and Y using $x_l + x_r = \frac{2FX}{Z}$ or $X = \frac{Z(x_l+x_r)}{2F} = \frac{B(x_l+x_r)}{2(x_l-x_r)}$. Using perspective projection model $y = \frac{FY}{Z}$. Hence $Y = \frac{yZ}{F} = \frac{yB}{x_l - x_r}$. Note that if you expressed X in other way (i.e., in terms of Z and x_l or x_r), it is OK as long as it is correct. (1pt for recovering Z, 0.5 pt for recovering X and Y, respectively.)

b) First we determine the location x_c of a 3D point that has x_l and x_r in the left and right image. Because $x_c = \frac{F\left(X+\frac{B}{4}\right)}{Z}$ based on the perspective projection, we have

$$x_c = \frac{F\left(X+\frac{B}{4}\right)}{Z} = x_l + \frac{F\left(-\frac{B}{4}\right)}{Z} = x_l + \frac{F\left(-\frac{B}{4}\right)(x_l - x_r)}{FB} = x_l - \frac{1}{4}(x_l - x_r) = \frac{3}{4}x_l + \frac{1}{4}x_r$$

(2 pt for correctly identifying the above relationship)
To generate the intermediate image, we can follow the following steps.

1) (2pt) Estimate the disparity of each pixel in the left image. That is, for each x_l, find its corresponding x_r. Then determine its corresponding location x_c in the intermediate view using the above equation.
2) Initialize the intermediate image to all zeros.
3) (4pt total) For each pixel in the left image (x_l, y), assign the pixel (x_c, y) a intensity value equal to a weighted average of the left image at (x_l, y) and the right image at (x_r, y). A reasonable choice of the weights could be ¾ for the left image and ¼ for the right image. But if you used the average, it is fine as well. (2pt)
Note that, it is possible that (x_c, y) may not correspond to an integer pixel location. A simple way to handle this is by quantizing x_c to the nearest integer position. (1pt) Also, (x_r, y) may not correspond to an integer position. Here instead of simply quantizing x_r to the nearest integer, it is better to interpolate the image value at (x_r, y) from the known values in integer positions using a chosen interpolation method, for example, the bilinear interpolation method. (1pt).

Ideally you should also note that sometimes you cannot find a corresponding pixel in the right image (i.e. due to occlusion). In that case, you need to have special treatment. This is not required.