Computer Vision (600.461/600.661)
Exam 1

Instructor: René Vidal

October 14, 2014

Part I (20 points) Answer these questions in 1-4 lines.

1. **(1 point)** Name two different color representations. Are they linearly related?

   ANSWER: RGB and YUV - yes these are linearly related **OR** RGB and HSV - no they are not linearly related

2. **(1 point)** Why is a Gaussian filter preferred to a box filter?

   ANSWER: A box filter often leads to image artifacts called ringing. The Gaussian filter does not do that.

3. **(1 point)** What do you do to sharpen an image?

   ANSWER: Scale the intensities of an image by 2 and then subtract from the result a smoothed version of the original image. Essentially, apply an impulse filter of amplitude 2, separately apply an averaging flter and subtract this from the former.

4. **(1 point)** What is the difference between the derivative of a Gaussian filter and the difference of Gaussians filter?

   ANSWER: The derivative of a Gaussian (w.r.t. x) is used to compute the derivative of the image (w.r.t. x) while at the same time smoothing with respect to the orthogonal direction, hence it can be used to detect edges. The difference of a Gaussian, on the other hand, is used to approximate the Laplacian of the image, hence it can be used to detect spots.

5. **(1 point)** What is template matching?

   ANSWER: Template matching is a technique for finding regions of an image that match a template. Template matching is usually done by convolving the image with a filter (the template) and looking for regions that give maximum response.

6. **(1 point)** What is a Gaussian pyramid? Name two applications of it.

   ANSWER: A Gaussian pyramid is a hierarchy of images. The bottom layer is an input image. The next layer is obtained by blurring the image in the previous layer and downsampling it, and so on. They are used in many computer vision algorithms such as SIFT, and optical flow estimation.

7. **(2 points)** List the main 3-5 steps of the Canny edge detector.

   ANSWER:

   (a) Filter the image I with the derivatives of Gaussians: I_{G_x} and I_{G_y}.

   (b) Find the magnitude and the orientation of the image gradient.

   (c) Find locations in the image where the magnitude of the gradient is above a threshold.

   (d) Perform non-maximum suppression to thin fat edges.

   (e) Perform the hysteresis thresholding to complete edges.

8. **(1 point)** Name any method described in class that requires image interpolation.

   ANSWER: Non-maximum suppression in Canny edge detection to find intensities along a line. Interest point localization in SIFT to obtain subpixel accuracy.

9. **(2 points)** List the main 3-5 steps of RANSAC as applied to line fitting with outliers.

   ANSWER:

(a) Randomly select 2 points on which to base the transformation estimate.

(b) Fit a line to the selected points.

(c) Find the points whose distance to the estimated line is smaller than a threshold (inliers).

(d) Terminate if the number of inliers or the number of iterations have exceeded a level. Otherwise repeat.

(e) Select the line with the largest number of inliers and re-fit a line using least squares considering all inliers.

10. **(2 points)** List the main 3-5 steps of the Harris corner detector.

    ANSWER:

    (a) Compute the image derivatives I_x and I_y by convolving the image I with the derivatives of a Gaussian filter $g(x, y)$.

    (b) Form the Harris matrix $H(x, y) = \sum_{u,v} g(u, v) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$.

    (c) Select points such that the Harris operator $K(x, y) = \det(H)/\operatorname{trace}(H)$ is above a threshold $K(x, y) > k$.

    (d) Perform non-maximum suppression.

11. **(2 points)** How do you make a patch descriptor rotationally invariant?

    ANSWER: One can make descriptors rotationally invariant by assigning orientations to the key points and then rotating the patch to a canonical orientation. In SIFT this is done by constructing Histograms of Gradients in a neighborhood around the feature point, and assigning the largest bin as the corresponding direction of the keypoint. Later, all detected features are rotated so that the corresponding orientations are vertically aligned.

12. **(1 point)** Name two criteria for deciding whether two feature descriptors match or not.

    ANSWER: Sum of Squared Differences: $SSD(d) = \sum_q \frac{(I_1(q) - I_2(q+d))^2}{n}$, Sum of Absolute Differences: $SAD(d) = \sum_q \frac{|I_1(q) - I_2(q+d)|}{n}$, or Normalized Cross Correlation: $NCC(d) = \sum_q \frac{\tilde{I}_1(q)\tilde{I}_2(q+d)}{\bar{\sigma_1}\bar{\sigma_2}(d)}$, where $\tilde{I}_1$ and $\tilde{I}_2$ are mean normalized values with $\bar{\sigma_1}$ and $\bar{\sigma_2}$ the corresponding standard deviations.

13. **(1 point)** How many point correspondences are needed to fit a 2D translational model?

    ANSWER: A 2D translation $(t_x, t_y)^\top$ has 2 degrees of freedom. Every 2D point correspondence $(x, y)^\top \leftrightarrow (x', y')^\top$ introduces 2 equations: $\begin{bmatrix} x + t_x \\ y + t_y \end{bmatrix} = \begin{bmatrix} x' \\ y' \end{bmatrix}$). Therefore, 1 set of point correspondences is enough to solve the 2D translational parameters.

14. **(1 point)** How many parameters (or degrees of freedom) are there in a 2D affine model?

    ANSWER: A 2D affine transformation involves 4 unknowns for the 2×2 linear affine transformation $A_{2\times2}$ and 2 unknowns for the 2D translation. In total, it has $4 + 2 = 6$ degrees of freedom.

15. **(1 point)** What is the aperture problem?

    ANSWER: The aperture problem refers to the fact that motion estimation is highly ambiguous when the observation window is very small.

16. **(1 point)** What is the direction in the image along which optical flow cannot be reliably estimated?

    ANSWER: The optical flow cannot be reliable estimated along the edges (perpendicular to the gradient direction). This is because the brightness constancy constraint involves the dot product between the image gradient ∇I and the optical flow $\boldsymbol{u}$, $\nabla I \cdot \boldsymbol{u} + I_t = 0$, which gives no equation on $\boldsymbol{u}$ when $\boldsymbol{u}$ and ∇I are orthogonal.

Part II (30 points) Solve the following problems.

1. **(15 points)** Suppose you have a large collection of photos from your trips, including photos of yourself alone, photos of yourself with other people, photos of other people without you, as well as photos without people. Suppose you also have a template of your face, which consists of a small image of your face together with the 2D coordinates for the center of your eyes, tip of your nose, and ears. Describe a fully automatic algorithm that

uses what you have learned in class to find the subset of the images that contains unoccluded frontal faces of you as well as the location, orientation and scale of a bounding box containing your face.

ANSWER: We provide three possible answers to the problem.

(a) Multi-Scale Multi-Orientation Template Matching: The simplest brute-force method is to apply the template at multiple locations, scales and orientations to the image and to compute a matching score. This will require applying upsampling and interpolation methods to the template and then convolving with the image. Then, we can select the location, scale and orientation that give the highest score. Now, since some images may not contain a face, it is important to verify if this score is above a threshold to make sure it is a proper detection. Moreover, since this templet matching procedure is expected to give high responses at the locations of other faces, the threshold needs to be high enough for images that contain faces of other people, but not your face.

(b) Multi-Scale Template Matching in an Image Pyramid + Orientation Estimation: A key disadvantage of the above approach is dealing with the computational complexity of having to evaluate the score function at every location, scale and orientation. To deal with this issue, it is preferable to keep the template as is and use an image pyramid with downsampled versions of the image to handle multiple scales. Now, an image pyramid only handles scale, but not orientation. Therefore, since the orientation of the template need not coincide with the orientation in the image, if you simply return the location and scale with the highest score, you may not get your face but rather someone else's face instead. Moreover, you would not be able to return the orientation of the face. To address this issue, we can keep the top 10-20 bounding boxes or return all bounding boxes above a certain threshold. If no bounding box is returned, your face is not present in the image. Then, at these 10-20 locations and scales, you can rotate the template to see if the score increases as we change the orientation. In this way, we have the best orientation for each one of the 10-20 bounding boxes. We can then select the bounding box with the best score and check if it is above a threshold, as before.

(c) Multi-Scale Template Matching in an Image Pyramid + Geometric Verification: In this approach, we also extract 10-20 bounding boxes using template matching on a pyramid, as before. But then we verify which detection is the correct one and estimate the orientation using a geometric verification procedure based on the extracted key points. More specifically, for each candidate bounding box, we find keypoints inside the box and extract SIFT features. We then match these with the SIFT features extracted from the template at the known keypoints. Given these matches, we can fit an affine model that maps the template to a candidate bounding box. This step is trying to find a face with the same structure (relative location of eyes, nose, ears) as the template. The bounding box which returns the best-fit affine model is declared as your face. To handle potential errors in point matching, for each bounding box we can generate several putative matches (e.g., top 5 scores in SSD matching), find an affine transformation for each case, and choose the best. Then, once an affine transformation has been found for each bounding box, we can extract the location and scale parameters from the bounding box location and scale. To find the orientation of the bounding box, you can extract the orientation parameter from the affine model as described later, or alternatively use the same procedure as in the previous method.

(d) Skin segmentation and verification using a geometric model: This is similar to method 2 described above, except that now instead of determining candidate bounding boxes, you use skin segmentation to determine faces like in HW2. In the ideal scenario, you will have a perfect segmentation of your face in the image, but as you saw in HW2, skin segmentation is not perfect. It will detect all skin-like regions in the image (arms, legs, etc) and also the skin regions of other people in the image. So, you repeat the geometric model verification part of method 2. You find keypoints in skin regions, extract SIFT features, match them to the SIFT features of known keypoints and then fit an affine model. The model that fits best if below a certain error threshold is your detected face. You can determine a bounding box by projecting the template onto the image using the affine model.

(e) Extracting location, scale and orientation parameters from the affine model: In two of the methods described above, we fit an affine model to determine your face in the image. We now describe how the model looks like and how to extract the location, scale and orientation parameters from it. An affine transformation is of the form $sR+T$ where s is the scale, R is the rotation matrix containing orientation parameter and T is the translation containing the location parameter. Assuming an affine transformation $A \in \mathbb{R}^{3\times 3}$, we know

that the structure is $A = \begin{bmatrix} sR & T \\ 0 & 1 \end{bmatrix}$ where $s \in \mathbb{R}^+$ and R is a rotation matrix of the form $\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$. So, simply extract s as the determinant of the 2×2 submatrix of A and extract orientation θ from R. T gives location parameter.

2. **(15 points) 3D affine registration.** Let $\{\boldsymbol{X}_i \in \mathbb{R}^3\}_{i=1}^N$ be a set of points in $\mathbb{R}^3$ that are transformed by a 3D affine transformation (A, T), where $A \in \mathbb{R}^{3\times3}$ and $T \in \mathbb{R}^3$, to produce another set of points $\{\boldsymbol{Y}_i \in \mathbb{R}^3\}_{i=1}^N$. Suppose that the transformed points $\boldsymbol{Y}_i$ are corrupted by noise $\boldsymbol{E}_i$, i.e., $\boldsymbol{Y}_i = A\boldsymbol{X}_i + T + \boldsymbol{E}_i$ for all $i = 1, \ldots, N$. Show that the transformation (A, T) that minimizes the sum of the squared errors

$$E(A, T) = \sum_{i=1}^{N} \|\boldsymbol{Y}_i - A\boldsymbol{X}_i - T\|_2^2 \tag{1}$$

is given by $T^* = \bar{Y} - A^*\bar{X}$, $A^* = (YX^\top)(XX^\top)^{-1}$, where $\bar{X} = \sum \boldsymbol{X}_i/N$, $X = \begin{bmatrix} \boldsymbol{X}_1 - \bar{X} \cdots \boldsymbol{X}_N - \bar{X} \end{bmatrix}$ and similarly for $\bar{Y}$ and Y. Show that 4 is the minimum number of points needed to find the transformation.

ANSWER: To minimize the cost E we set the first derivative to zero as:

$$\frac{\partial}{\partial T} E(A, T) = -2\sum_{i=1}^{N}(\boldsymbol{Y}_i - A\boldsymbol{X}_i - T) = 0 \implies T^* = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{Y}_i - A\boldsymbol{X}_i) = \bar{Y} - A^*\bar{X} \qquad \text{(3 points).}$$

Now, we can substitute the translation in the cost function and obtain:

$$E(A) = \sum_{i=1}^{N} \|\boldsymbol{Y}_i - \bar{Y} - A(\boldsymbol{X}_i - \bar{X})\|_2^2 = \|Y - AX\|_F^2. \qquad \text{(3 points)}$$

The minimization problem of $\min_A \|Y - AX\|_F^2$ is answered by taking the derivative with respect to A and setting it to zero:

$$-(Y - AX)X^\top = 0 \implies A^* = (YX^\top)(XX^\top)^{-1} \qquad \text{(6 points)}$$

A general 3D affine transformation introduces 3 DOF (Degrees of Freedom) for the $T \in \mathbb{R}^3$ and 9 DOF for the linear affine matrix $A \in \mathbb{R}^{3\times3}$. Therefore, there are 12 parameters to estimate. Every 3D point correspondence gives 3 sets of equations. Therefore, at least 4 point correspondences are required ($4 \times 3 = 12$) for a unique solution over the 12 parameters. (3 points)