CS4670 Spring 2015

HW2 - Reconstruction, Photometric Stereo, Recognition

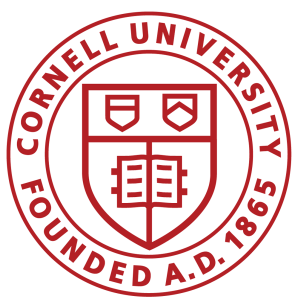
Due: Apr 29 (Wed), 2015

Homeworks to be done alone. Remember the academic integrity policies.
Consult piazza for clarifications on the HW2 FAQ.
Remember to show your work for all problems.

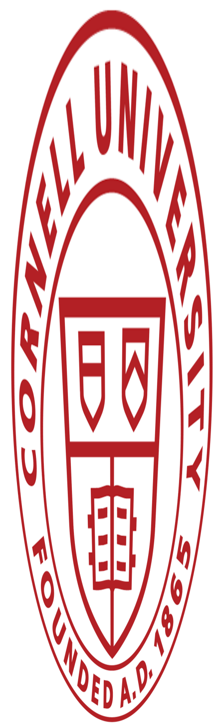
1 Optical illusion

In this problem you have to create an optical illusion. The scene configuration is as follows. There is a camera (center of projection) at $(0, 0.4, 0)$, the floor with normal vector $(0, 1, 0)$, going though the origin, a billboard in vertical position on the floor with points $A = (-0.1, 0, 1), B = (0.1, 0, 1), C = (-0.1, 0.2, 1), D = (0.1, 0.2, 1)$. Your task is to compute the image which should be drawn on the floor to provide the same picture as the vertical billboard from the viewpoint of the camera. There are examples for optical illusion art on the internet (link). See an example below (1b and 1c):

(a) Illusion, Michael!



(b) Image splatted on the billboard



(c) Transformed image, which should be put on the floor.

1.1 Compute the projection of the corners of the billboard to the floor using the camera as the center of projection. Write down the coordinates of the projected points in your answer.

1.2 Now imagine that we tip down the billboard on the floor, but we keep the two points which were already on the floor at the same position. Write down the 2D coordinates of the points of the tipped down billboard and the projected billboard in the plane of the floor. The floor plane coordinate system's x axis is the same as the original 3D coordinate system and the y axis is the same as the original 3D coordinate system's z axis, but it is 0 at points A, B.
Compute the homography to transform the points of the billboard to the projected billboard points in the plane of the floor using the 4-point algorithm discussed in lecture (link to lec 12, slide 27). You can use your implementation from pa3. Write down the normalized homography in your answer (the bottom-right element of the matrix should be 1).

1.3 Use the computed homography to transform the provided image (1a) and print it. You first need to transform from image coordinates to billboard coordinates (you computed these coordinates in 1.2), let's call this transformation T. Then apply your computed homography and finally transform back to image coordinates using T^{-1}. You can use OpenCV's perspectiveTransform function to compute the corners of the transformed image and warpPerspective for warping the image itself.

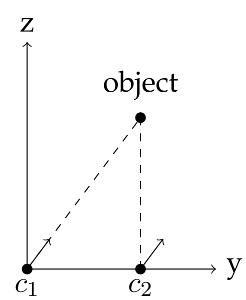
1.4 If you shut one of your eyes and move to the right position, you should see the optical illusion! Include the original and the transformed image, and a photo (taken from the right location and angle) to showcase the illusion with your submission.

2 Structure from Motion

There are two cameras taking pictures of an object, as depicted in the figure below. We want to compute the location of the object.

- Camera 1 has focal length 2 and Camera 2 has focal length 4. Assume no other intrinsics are in play.
- Camera 1 has center position $c_1 = (0, 0, 0)$ and Camera 2 has center position $c_2 = (0, 3, 0)$. The arrows leaving c_1 and c_2 in the figure show the directions in which the cameras are pointed.
- Both Camera 1 and Camera 2 have 4×4 rotation matrix R, given below. Note that R takes orientations in world coordinates to orientations in camera coordinates. Remember that, in camera coordinates, the $-z$ axis is the direction in which the camera points. (04/07/2015)

$$R = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -0.8 & 0.6 & 0 \\ 0 & -0.6 & -0.8 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



2.1 The image coordinates of the object on the image from Camera 1 are $(0, 0)$. Using these image coordinates, find the vector pointing from the center position of Camera 1 to the point on the image plane corresponding to the object, with respect to the camera's 3-D coordinate system.

2.2 Find the vector pointing from the world origin $(0, 0, 0)$ to the point on the image plane corresponding of the object, with respect to the world's coordinate system.

2.3 Using the vector obtained in (2.2) and Camera 1's center position, find the equation of the line through Camera 1's center and the object, with respect to the world's coordinate system. Provide an equation in parametric form that's defined for non-homogeneous 3-D points.

2.4 The image coordinates of the object on the image from Camera 2 are $(0, 3)$. Using the steps in parts (2.1) through (2.3), find the equation of the line through Camera 2's center and the object.

2.5 These lines intersect at the location of the object. Compute the object's location.

3 Photometric Stereo

Suppose we have a Lambertian surface point P with albedo 1.0 at 3D coordinate $\begin{bmatrix}0 & 0 & 0\end{bmatrix}^T$, viewed by a camera at $\begin{bmatrix}0 & 0 & 1\end{bmatrix}^T$ (looking towards the -z axis). Suppose P projects to a 2D pixel location p. We'll consider three images taken by this camera, with 3 light sources illuminating the surface. Assume all light sources are directional (infinitely distant point lights), and suppose that their directions from the origin are:

- $\ell_1 = \frac{1}{\sqrt{3}} * \begin{bmatrix}1 & 1 & 1\end{bmatrix}^T$
- $\ell_2 = \frac{1}{\sqrt{3}} * \begin{bmatrix}-1 & 1 & 1\end{bmatrix}^T$
- $\ell_3 = \begin{bmatrix}0 & 0 & 1\end{bmatrix}^T$

Assume the camera's response function is linear and light source brightnesses are calibrated so that the maximum possible pixel value is 1.0.

3.1 In image 1, illuminated by ℓ_1, the pixel value at p is 0.11547. Given only this information, how many possible values are there of the surface normal direction?

3.2 Suppose that the surface normal $\vec{n}$ is $\begin{bmatrix}0 & -0.6 & 0.8\end{bmatrix}^T$. What are the pixel values of p in images 2 and 3?

3.3 Which of the following could change the observed intensity value of a point on the surface?

- Translating the surface
- Rotating the surface
- Translating the camera
- Rotating the camera
- Changing the albedo of the surface

3.4 The camera above is now taking an orthographic image of a chrome ball centered at the origin with radius 20, lit by a single directional light source. The highlight is located at pixel coordinates $(4, 10)$. What is the surface normal at this point?

3.5 Compute the unit vector pointing from the origin towards the light source.

4 CNNs

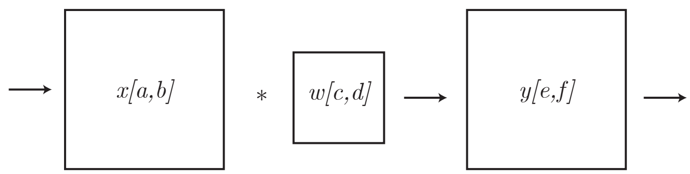
Background. In this problem, we will derive an update rule for a convolutional neural network (CNN). A neural network consists of a number of layers, each layer having parameters w. One type of layer is a "convolutional" layer that convolves an input x with a set of weights w to produce an output y:

$$y = x \star w \tag{1}$$

Here, x is a 2D grayscale image, w is a 2D array of weights, and y is an output 2D grayscale image. We will assume zero padding for all convolution. We can write this as an equation (the definition of convolution):

$$y[e, f] = \sum_a \sum_b x[a, b] \cdot w[e - a, f - b] \tag{2}$$

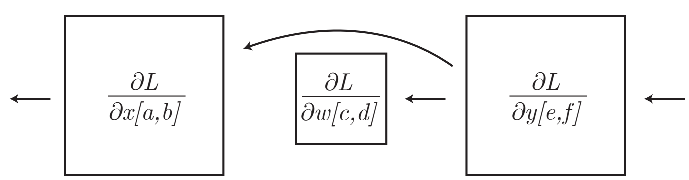
or we can write it as a block diagram:



The block diagram is useful to visualize the relative sizes of each layer. To avoid confusing indices, we will index the input as $x[a, b]$, the weights as $w[c, d]$, and the output as $y[e, f]$.

Backpropagation. To "learn" the weights for a CNN, it is necessary to compute the gradient of some loss function L with respect to the weights inside the network. L measures "how good" the current weights are, and we can use the gradient $\frac{\partial L}{\partial w}$ to update the weights to be slightly better (gradient descent).

The goal of backpropagation is to calculate $\frac{\partial L}{\partial x[a,b]}$ and $\frac{\partial L}{\partial w[c,d]}$ in terms of $\frac{\partial L}{\partial y[e,f]}$, where L is some function that depends on y (and y depends on x and w). Note that x is a 2D image, so $\frac{\partial L}{\partial x}$ is a 2D image of the same size (and similarly for w and x). We can write this as a block diagram to see the sizes of each image:



Note that we don't need to know L, because we can use the chain rule to relate partial derivatives together:

$$\frac{\partial L}{\partial x[a, b]} = \sum_e \sum_f \frac{\partial L}{\partial y[e, f]} \cdot \frac{\partial y[e, f]}{\partial x[a, b]} \tag{3}$$

Computing y is called the "forward" pass (because the computation proceeds left to right) and computing $\frac{\partial L}{\partial x}$ and $\frac{\partial L}{\partial w}$ is the "backward pass" because it proceeds right to left.

4.1: Convolution layer.

1. For a convolution layer that computes

$$y = x \star w$$

Show that the backwards pass is the same as the forwards pass, but with a weight kernel that is flipped vertically and horizontally. Specifically, derive the gradient with respect to the input x and show that it is:

$$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial y} \star \tilde{w}$$

where $\tilde{w}$ is a flipped version of w both vertically and horizontally: $\tilde{w}[c, d] = w[-c, -d]$.

Hint: start from Equation 3.

2. Similarly, show that the gradient with respect to the weights w is also a flipped convolution:

$$\frac{\partial L}{\partial w} = \frac{\partial L}{\partial y} \star \tilde{x}$$

where $\tilde{x}$ is a flipped version of x: $\tilde{x}[a, b] = x[-a, -b]$.

Hint: remember that $x \star w = w \star x$.

4.2: Fully connected layer.

1. Consider another type of layer that computes matrix multiplication:

$$y = Wx$$

where x is an input 1D vector, W is a 2D weight matrix, and y is an output 1D vector. We call this "fully connected" because every entry in y depends on every entry in x.

Derive the backward pass for this layer and write your answer in the form:

$$\frac{\partial L}{\partial x} = A^T B$$

Hint: convert all matrix expressions to operations on scalars (e.g., write out the definition of the matrix product), manipulate everything as scalars, and then convert back to matrix expressions in the last step.

Hint: here is the chain rule for $\frac{\partial L}{\partial x}$:

$$\frac{\partial L}{\partial x_i} = \sum_j \frac{\partial L}{\partial y_j} \frac{\partial y_j}{\partial x_i}$$

2. Similarly, derive the gradient with respect to the weights W and write your answer in the form:

$$\frac{\partial L}{\partial W} = AB^T$$

Hint: write down an expression for $\frac{\partial L}{\partial w_{ij}}$ and start from there.

Note: An answer with no derivation will receive no credit.