Assignment 3: CS 763, Computer Vision

Sasank Chilamkurthy
Tharun Kumar Reddy
Rajeev Puppala

March 23, 2015

1. Consider a surface with Lambertian reflectance map, known geometry (example: sphere) and unknown but constant albedo. Given an image of such a surface taken with a point light source of unknown power, show how you will determine the lighting direction. Assume there are no shadows. Write down all necessary equations. [3 points]

Ans. The scene Irradiance is given by equation

$$I(x, y) = L\rho\mathbf{l}^t N(x, y) \tag{1}$$

$I =$ scene Irradiance
$L =$ lighting intensity
$l = (-l_x, -l_y, l_z)$ lighting direction
$N(x, y) =$ surface normal at that point
$\rho(x, y) =$ surface reflectance(albedo)
reflectance map and geometry($N(x, y)$ at all points) are known but L and ρ are unknown.
Take three points $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ and evaluate the above equation at these points.
Taking ratio removes the constant $L\rho$ from the list of unknowns. Now we have three unknowns $(-l_x, -l_y, l_z)$ and three equations to solve them.

In another method we can look for the maximum scene Irradiance from the reflectance map and the Normal at that point will give the lighting direction.

2. A Lambertian object illuminated by a point source has a reflectance map of the form given by

$$R(p, q) = \frac{1 + pp_s + qq_s}{\sqrt{1 + p_s^2 + q_s^2}\sqrt{1 + p^2 + q^2}} \tag{2}$$

where the surface normal is $(-p, -q, 1)$ and the light source direction is $(-p_s, -q_s, 1)$. What value(s) of (p, q) will maximize $R(p, q)$? For what values, will you get $R(p, q) = 0$? [2 points]

Ans. Scene Irradiance is maximum when the normal at a given point is parallel to the lighting direction and Zero when it is perpendicular to lighting direction
surface normal is $(-p, -q, 1)$ and the light source direction is $(-p_s, -q_s, 1)$.

(a) $R(p, q)$ will be maximum if $(-p, -q, 1)$ is parallel to $(-p_s, -q_s, 1)$
Therefore $p = p_s$ and $q = q_s$

(b) $R(p, q) = 0$ if $(-p, -q, 1)$ is perpendicular to $(-p_s, -q_s, 1)$
$\implies pp_s + qq_s + 1 = 0.$
$\implies pp_s + qq_s = -1$
$\implies (p, q)$ lies on a line formed by the above equation.

3. Consider K images of a stationary Lambertian object captured from an orthographic camera at a fixed viewpoint, under K different lighting directions and lighting intensities respectively. For the i^{th} lighting condition, we write $I_i(x,y) = L_i\rho(x,y)\mathbf{N}^t(x,y)\mathbf{d_i}$, where $1 \leq i \leq K$, $\mathbf{d_i}$ is a lighting direction, L_i is the power of the light source, $\rho(x,y)$ is the albedo and $\mathbf{N}(x,y)$ is the unit surface normal vector at pixel (x,y) (we will assume that there were no shadows at any pixel in any image). We can combine all K equations to write the relation $\mathbf{I} = \tilde{\mathbf{N}}\tilde{\mathbf{D}}$. Here $\mathbf{I} \in \mathbb{R}^{M\times K}$ is a matrix whose i^{th} column $(1 \leq i \leq K)$ contains the i^{th} image in vectorized form. $\tilde{\mathbf{N}} \in \mathbb{R}^{M\times 3}$ is a matrix whose j^{th} row contains the product of the unit surface normal vector at some point and the albedo at that point, i.e. $\tilde{\mathbf{N}}_j$ (the j^{th} row of $\tilde{\mathbf{N}}$) is given by $\tilde{\mathbf{N}}_j = \rho(x,y)\mathbf{N}(x,y)$ where (x,y) will go to row j when the image is vectorized. $\tilde{\mathbf{D}} \in \mathbb{R}^{3\times K}$ is a matrix whose i^{th} column contains $L_i\mathbf{d_i}$. Now answer the following:

(a) Show that $\mathbf{I}$ has rank 3 in the absence of noise. [2 points]

Ans. $\mathbf{I} = \tilde{\mathbf{N}}\tilde{\mathbf{D}}$

$\tilde{\mathbf{N}} \in \mathbb{R}^{M\times 3}$ and $\tilde{\mathbf{D}} \in \mathbb{R}^{3\times K}$

Assuming that $M >= 3$ and $K >= 3$

Rank of $\tilde{\mathbf{N}} = 3$ and Rank of $\tilde{\mathbf{D}} = 3$

Therefore Rank of $\mathbf{I} = 3$ in the absence of noise.

(b) We have seen (or will soon see) the photometric stereo problem in class where we estimate surface normals from the images under different lighting conditions, assuming the lighting directions were known. Now suppose, we did not know the lighting directions $\{\mathbf{d_i}\}$, the light source intensities $\{L_i\}$, the albedo at each point on the surface of the object $\{\rho(x,y)\}$, and the surface normals $\{\mathbf{N}(x,y)\}$. Given $\mathbf{I}$, we can perform the SVD to 'find' $\tilde{\mathbf{N}}$ and $\tilde{\mathbf{D}}$. But the decomposition will be unique only up to an unknown 3×3 invertible transformation $\mathbf{A}$. (There is a very interesting similarity between this problem and the Tomasi-Kanade factorization in structure from motion.) Now, consider that the albedo at some $m > 1$ points was known. Show how this information can help you make the decomposition unique up to an unknown orthonormal transformation $\mathbf{R}$. What is the minimum m needed? What will happen if you didn't know the actual albedo values at these m points, but only knew that they were all equal? [4 points]

Ans. Computing SVD of $\mathbf{I}$ we get

$\underset{M\times K}{I} = \underset{M\times M}{U}\underset{M\times K}{S}\underset{K\times K}{V^T}$

Reduced form of SVD as Rank of $\mathbf{I} = 3$ is

$\underset{M\times K}{I} = \underset{M\times 3}{U}\underset{3\times 3}{S}\underset{K\times 3}{V^T}$

$\implies \underset{M\times K}{I} = \underset{M\times 3}{U}\underset{3\times 3}{S^{\frac{1}{2}}}\underset{3\times 3}{S^{\frac{1}{2}}}\underset{K\times 3}{V^T}$

Therefore $\tilde{\mathbf{N}} = \underset{M\times 3}{U}\underset{3\times 3}{S^{\frac{1}{2}}}$ and $\tilde{\mathbf{D}} = \underset{3\times 3}{S^{\frac{1}{2}}}\underset{K\times 3}{V^T}$

But they are not unique because for any 3×3 invertible matrix $\mathbf{A}$ we can write

$\mathbf{I} = \tilde{\mathbf{N}}\mathbf{A}\mathbf{A}^{-}\tilde{\mathbf{D}}$

Rows of $\tilde{\mathbf{N}}$ have magnitude of ρ at that given point.

$\mathbf{A}$ has degrees of freedom = 9.To solve this using Newton's method we need 9 independent equations. Albedo(ρ) at a point will give one equation. So albedo should be known at atleast 9 different points to solve for $\mathbf{A}$

$\therefore m >= 9$

If it is given that albedo at m points are equal instead of actual albedo values they give only $m-1$ independent equations by equating the two equations formed by two different points. Therefore atleast 10 points should be given to solve for $\mathbf{A}$

$\therefore m >= 10$

(c) Instead of marking out points with equal albedo, suppose you were told that the intensity of the light source in $m > 1$ images was known. Show again how this information can help you make the decomposition unique up to an unknown orthonormal transformation $\mathbf{R}$. What is the minimum m needed? What will happen if you didn't know the actual intensities, but only knew that they were all equal? [4 points]

Ans. Same as the previous case but instead of rows in $\tilde{\mathbf{N}}$ we take columns in $\tilde{\mathbf{D}}$. The magnitude of columns is L_i which implies that magnitude of rows in $\tilde{\mathbf{D^T}}$ are L_i
Therefore to solve for $\mathbf{A}$ which has 9 degrees of freedom we need 9 independent equations. L_i for a particular lighting condition gives one equation and 9 such equations are necessary
$\therefore m >= 9$
If it is given that lighting intensities of m images are equal instead of actual intensities they give only $m-1$ independent equations by equating the two equations formed by two different images. Therefore atleast 10 points should be given to solve for $\mathbf{A}$
$\therefore m >= 10$

4. In this exercise, you will implement a software routine to stabilize a video. Videos acquired by handheld cameras or smartphones often appear jerky or shaky due to inevitable motion of the hand during acquisition. This artifact is exacerbated in videos acquired from handheld devices while inside a moving vehicle. The processing of removing the unwanted motion between consecutive frames (while maintaining or preserving the intended motion) is called as video stabilization. In some papers, video stabilization also comprises removal of motion blur, but we will not consider that in this exercise. Your task here is as follows:

- The videos can be read in MATLAB using the 'mmread' routine from the package. The routine supports .mp4, .mpeg, .avi formats besides others. Generate shaky versions of these videos using the following MATLAB routines provided in the homework folder: 'generate_shaky_video_TranslationOnly.m' (for 2D translations), 'generate_shaky_video_Rigid.m' (for 2D translations + in-plane rotations) or 'generate_shaky_video_Affine.m' (for affine motion in 2D). These routines assume the first frame of the video is already 'stable' and apply random motion only to the subsequent frames. Also, the folder contains a MATLAB function called 'displayvideo.m' which takes a video in the form of a 3D array and displays it at a specified rate, and another function called 'writeVideo' which writes a video (in the form of 3D array) to an uncompressed .avi file with a specified frame rate.
- You now have to estimate the motion between frames n and $n-1$ $(1 < n \leq T)$ of the shaky video. For this, you should use the SIFT algorithm to (1) detect salient feature points in both the frames and (2) determine matches in between the points of those frames. Now, given this set of matching pairs of points produced by the SIFT package, your first task is to estimate the motion in between them - which will (hopefully) be the same as the motion between the frames. You should perform this using two methods: (1) Least squares, and (2) RANSAC (which we are currently studying in class). You should repeat this for all pairs of consecutive frames and generate a motion sequence. A motion sequence is a sequence containing the motion parameters at every frame. For instance, assuming a translation+rotation model, the motion sequence acquires the form $\{(t_x^{(n)}, t_y^{(n)}, \theta_x^{(n)})\}_{n=2}^{T}$. [3 points - 1 point each for pure translation, translation + rotation, and affine]

Ans. We fitted a homography which exaplains all of the pure translation, translation + rotation, and affine models and more. So our motion model is of the form

$$\{H_i\}_{i=1}^{N-1}$$

where H_i is homography estimated between ith and $i+1$th frame.

- For a shaky video, this motion sequence will be very noisy. You should smooth the sequence using a simple averaging filter to generate a smoothed motion sequence. The width of the averaging filter is a user-choice. Make sure your averaging filter is wide enough or the amount of smoothing may be inadequate. Plot two examples of noisy and smoothed sequences in your report for every motion model you experiment with. [1 point]

Ans. We plot $\{t_x\} = \{H_i(1,3)\}_{i=1}^{N-1}$ and $\{t_x\} = \{H_i(1,3)\}_{i=1}^{N-1}$. Smoothing is done using a Gaussian filter.

- Now given the smoothed sequence, re-warp the different frames of the shaky video to generate a more stable version of the video. (My hunch is that most of you will find this part to be trickier than what is may initially appear!). View the shaky and stabilized videos together by (1) putting them in a single array of size $2H \times W \times T$ where (H, W) is the size of each frame and T is the number of frames, and

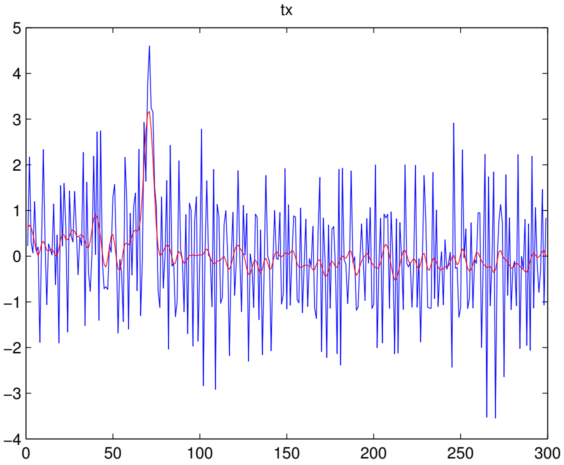

Figure 1: $H_i(1,3)$ vs i

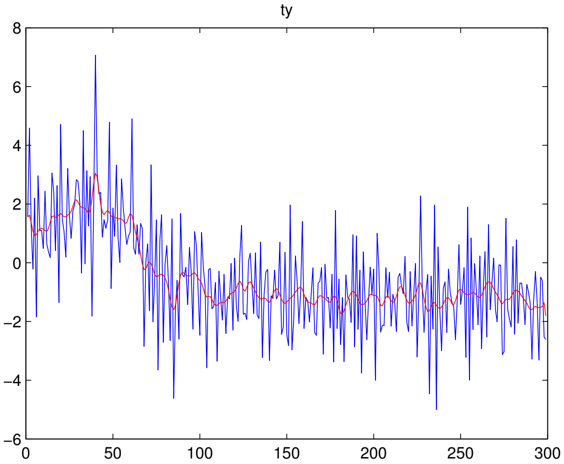

Figure 2: $H_i(2,3)$ vs i

(2) using the routine ‘displayvideo’ mentioned earlier. Also your MATLAB program should write the combined video to a file and give it a logical name that you print on screen. [5 points for successfully dealing with 2D translation + 3 points for rotation + 2 BONUS points for the affine case]

Ans. Because of the limitations of the moodle and as per instruction of professor on moodle, All but one video from the submission folder are removed. Complete video results are made available at `https://github.com/chsasank/CV/tree/master/HW3/Video%20Stabilization/results`.

- After patting yourself on the back for your hard work :-), it’s now time to act as your own critic: What are the limitations of what you have developed? For example, can you think of certain types of videos or motion models or other situations your program is or will be unable to handle? [3 points]

Ans. If the underlying motion of the video can’t be explained by a homography, our method fails badly. An example can be seen in last few frames of ’gbus.avi’. Here, different vehicles move at different speeds making it impossible to be explained by a homography.