# EE5175: Image Signal Processing

# Lab-5 (Optional)

Motion blur

## A Simulated motion blur

In this programming assignment, we will study how to generate a motion blurred image from the given camera motion trajectory.

1. **Pose-space modeling of camera motion** There are two ways to generate the blurred image; one is to average all the warped images due to individual poses over exposure time, and another is to perform weighted averaging of warps based on the time spent by the camera at each pose.

   (a) Use the given camera trajectory ('cam.txt') and the clean image **I** ('pillars.jpg') to generate a sequence of images corresponding to the camera motion. Average all the generated images to produce a blurred image ($\mathbf{B}_1$). Assume the normal of the scene $\mathbf{n}$ = [0 0 1]$^T$, the depth of the scene $d$ = 1000 pixels, and the focal length of the camera $f$ = 500 pixels.

   (b) Find out a set ($S$) of unique camera poses and corresponding weights ($\mathbf{W}$) (the number of times the camera goes through a pose during motion / total number of poses). Use each pose from $S$ to generate corresponding warped images and perform a weighted averaging (using $\mathbf{W}$) to produce a blurred image ($\mathbf{B}_2$).

   Verify that the blurred images ($\mathbf{B}_1$) and ($\mathbf{B}_2$) are same by calculating the mean square error (MSE($\mathbf{B}_1$,$\mathbf{B}_2$)) between them.

$$MSE(\mathbf{B}_1, \mathbf{B}_2) = \frac{1}{N_{pix}} \sum_{i,j} (\mathbf{B}_1(i,j) - \mathbf{B}_2(i,j))^2 \tag{1}$$

   where $N_{pix}$ is the total number of pixels in $\mathbf{B}_1$ (or $\mathbf{B}_2$).

2. **Effect of depth on blur induced by in-plane translations ($t_x$, $t_y$)**
   Generate blurred image (using 1(a)) with in-plane translations ($t_x$, $t_y$) alone (keeping other camera motion values as zero). Vary the depth ($d$) of the scene (say $d$ = 200, 1000, 5000), and observe how the blur varies for each case. Assume $\mathbf{n}$ = [0 0 1]$^T$, $f$ = 500 pixels.

3. **Effect of focal length on blur induced by out-of-plane rotations** ($r_x$, $r_y$)
   Generate blurred image (using 1(a)) with out-of-plane rotations ($r_x$, $r_y$) alone. Vary the focal

length ($f$) of the camera (say $f$ = 100, 500, 1000), and observe how the blur varies for each case. Assume $\mathbf{n} = [0\ 0\ 1]^T$, and $d$ = 1000 pixels.

**For all above experiments, to visualize the effect of camera motion**

do

either

(i) Generate a 'gif' image using the images corresponding to all the poses (use 'imwrite' with 'gif' options in MATLAB).

or

(ii) Display the images corresponding to all poses consecutively over the same figure window.

## Steps

Camera trajectory contains a set ($A$) of continuous 6D camera poses ($t_x, t_y, t_z, r_x, r_y, r_z$) corresponding to a camera motion.

**A) To generate motion blurred image as per 1(a)**

(i) Find the homography corresponding to each pose from the camera trajectory using the following equation,

$$\mathbf{H} = \mathbf{K}(\mathbf{R} + \frac{\mathbf{t}\mathbf{n}^T}{d})\mathbf{K}^{-1} \tag{2}$$

where $\mathbf{K} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}$, $\mathbf{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$, and $\mathbf{R} = \mathbf{R}_z(r_z).\mathbf{R}_y(r_y).\mathbf{R}_x(r_x)$.

$$\mathbf{R}_z(r_z) = \begin{bmatrix} \cos r_z & -\sin r_z & 0 \\ \sin r_z & \cos r_z & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{R}_y(r_y) = \begin{bmatrix} \cos r_y & 0 & \sin r_y \\ 0 & 1 & 0 \\ -\sin r_y & 0 & \cos r_y \end{bmatrix},$$

$$\mathbf{R}_x(r_x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos r_x & -\sin r_x \\ 0 & \sin r_x & cos r_x \end{bmatrix}$$

(ii) Apply the homographies ($\mathbf{H_p}$) (obtained from poses ($\mathbf{p}$) in camera trajectory ($A$)) on the clean image ($\mathbf{I}$) to produce the images corresponding to different poses of the camera and average all of them to produce the blurred image as follows.

$\mathbf{B}_1 = \frac{1}{N_{total}} \sum_{\mathbf{p} \in A} (\mathbf{H}_p \circ \mathbf{I})$, where $\mathbf{H_p} \circ \mathbf{I}$ refers to the application of homography warp on $\mathbf{I}$, and $N_{total}$ is the total number of poses in the camera trajectory $A$.

**B) To generate motion blurred image as per 1(b)**

(i) Find the set of unique poses ($S$) among the collection of all poses ($A$) in the camera trajectory (where $S \subseteq A$). Find the number ($N_\mathbf{p}$) of times each pose ($\mathbf{p}$) is repeated within the camera motion. Find the weight ($W_\mathbf{p}$) for each pose as, $W_\mathbf{p} = \frac{N_\mathbf{p}}{N_{total}}$

(ii) Find the warped images ($\mathbf{H_p} \circ \mathbf{I}$) corresponding to each pose $\mathbf{p} \in S$, and perform the following weighted averaging to produce the blurred image.

$\mathbf{B}_2 = \sum_{\mathbf{p} \in S} W_{\mathbf{p}}(\mathbf{H}_{\mathbf{p}} \circ \mathbf{I})$

**C) To read the camera trajectory from the file**

Use the input 6D camera pose values corresponding to $t_x, t_y, t_z, r_x, r_y$, and $r_z$ provided in 'cam.txt'. Each camera pose ($\mathbf{p} \in A$) can be formed as

$\mathbf{p} = (t_x(i), t_y(i), t_z(i), r_x(i), r_y(i), r_z(i))$ for $i = 1, .., N_{total}$.

The translational values are specified in pixels, while the rotation values are in radians.

# B Realistic Motion Blur

In the file `frames.mat`, you are provided with multiple frames captured sequentially using a high speed camera over a time-duration T.

1. How can one synthesize a blurry image by directly employing the image-stack? Create a blurry image with exposure duration T.
2. Create blurry image indirectly: i.e., consider the middle image of the stack as reference and compute the homographies for each image with respect to the reference image (using RANSAC code used for mosaicing). Apply the estimated homographies on the reference image to synthesize the image-stack and create blurry image using the synthesized image-stack.
3. Compare the blurry images and image-stacks of part (1) and (2). Also, compute the error between individual image-pairs of the two stacks, and plot the errors.
4. What could be the possible factors that lead to these errors?

–end–