

HW6: Synthetic Aperture Imaging (15 points)

Due: Thursday 12/3 at 11:59pm

EECS 395/495: Introduction to Computational Photography

In this homework, you will capture a light field by waving your tablet camera around. The goal of this homework is to synthesize an image that appears to be captured from a synthetic aperture that is much larger aperture than your tablet camera [1]. The advantage of this technique is that captured images can have a significantly smaller depth of field, giving the look and feel of a much more expensive camera. The other advantage is that, because we are capturing a light field, we can digitally refocus the image. The disadvantage of this technique is that, because it relies on sequential image capture, it will only work for static scenes. The technique requires the image sequence to be registered, and the quality of the final photograph will be limited by how well the images can be registered. In this homework, we will use a simple auto-correlation based registration scheme.

1. Capture an unstructured light field (3 points)

You don't need to write any Android code for this homework assignment. You can use the any Android camera application that records video. You will capture a video as you wave your camera around in front of a static scene. Here are some guidelines.

1. You need to avoid tilting and rotating the camera as much as possible. You should just shift the camera in a plane (see Figure 1). You may need to try this several times before you capture a good video.
2. You may want to press the phone against a flat object to help guide the planar motion
3. Experiment with different types of camera motion. You can try moving the camera in a zig-zag motion (as in Figure 1). You can also try using circular motion. In general, the more you cover the plane, the better your results in the next section will be.
4. Make sure your video is not too long. You should capture a few seconds of video.
5. Capture a scene with a few objects at different depths. See Figure 2 for an example. In the next section you will create narrow synthetic aperture images that are focused on these objects
6. Make sure that all of the objects in the scene are in focus. As long as objects are about a meter in size and 1-2 meters away, they will more or less be in focus.

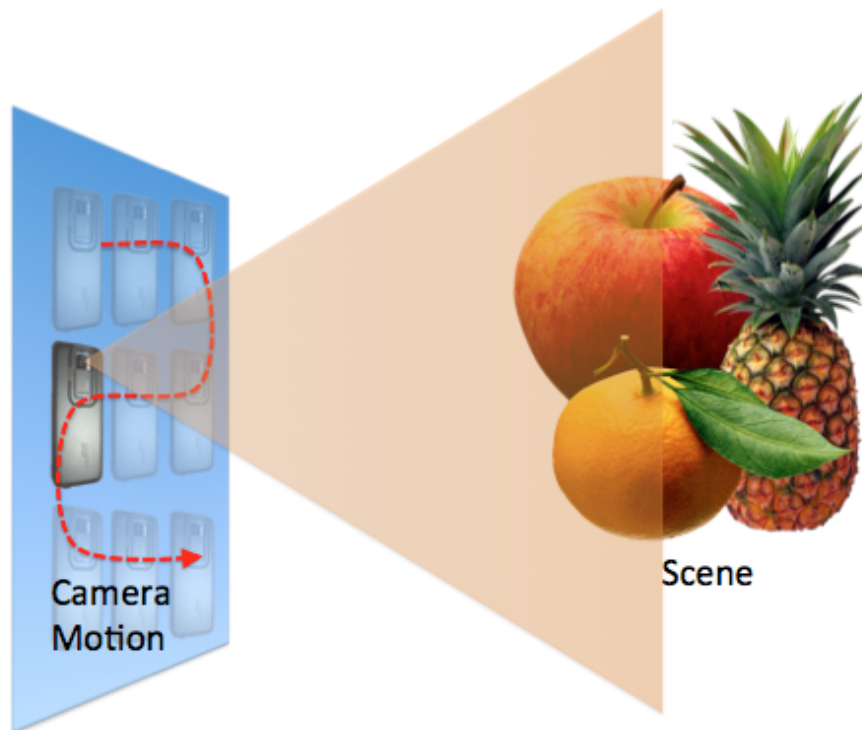


Figure 1: Capture an unstructured light field of a scene by waving a camera in front of a scene. Make sure the motion is in a plane.

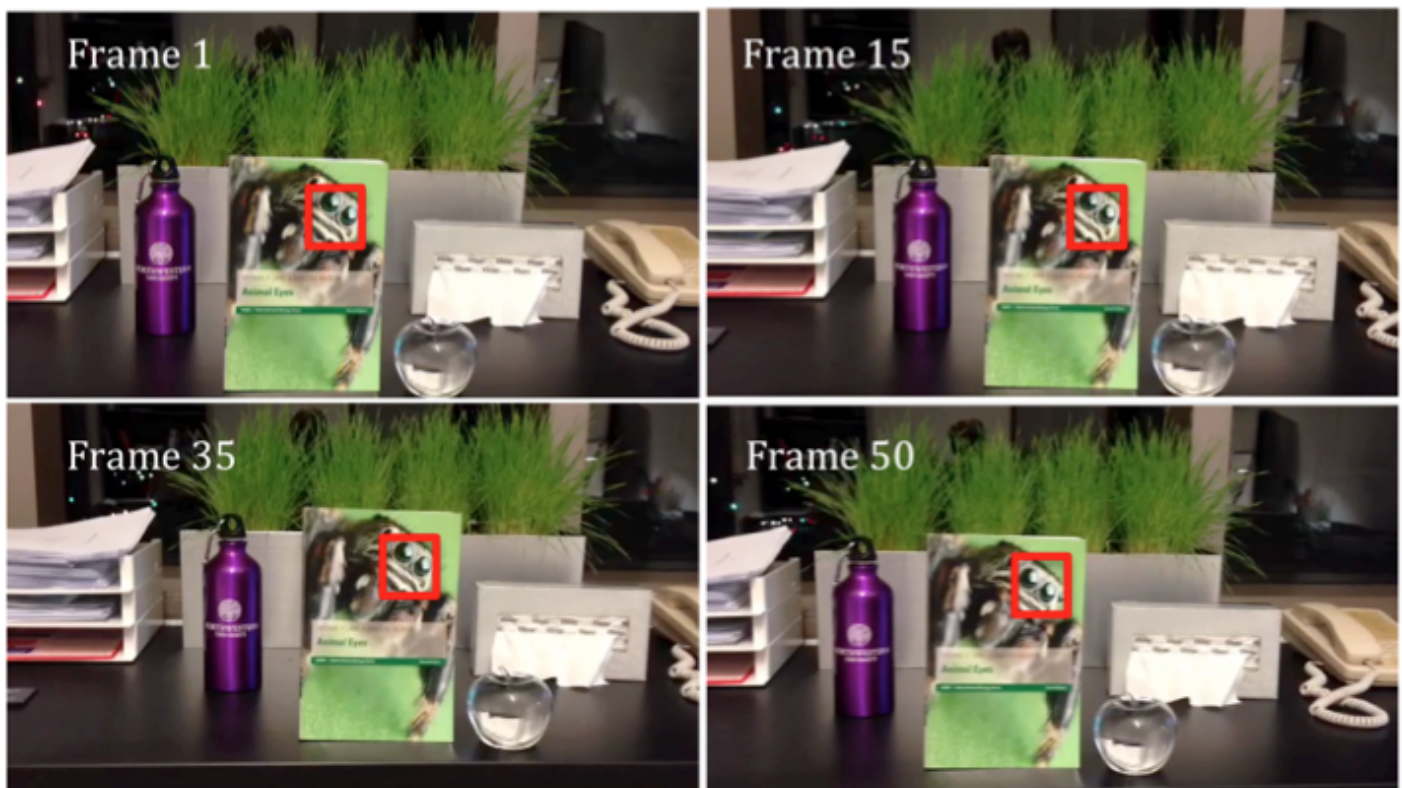


Figure 2: four images from a video captured using the zig-zag motion shown in Figure 1. The red box shows the template that is used to register the video frames in the next part 2 of this homework.

2. Register the frames of video using template matching (5 points)

Once you have captured a video, you use Matlab to register each frame of captured video and generate synthetic aperture photos. *Registration should be performed on grayscale*

video frames. You will use a simple template-matching scheme as described in the first lecture on image processing. Here is what you need to do:

1. Write a matlab program to load in the video. You can use the built-in Matlab VideoReader object to read the saved .mp4 file into a Matlab array (see documentation for details).
2. Convert each frame of your video to grayscale
3. You can use all of the frames in the video, but this may make your processing slow if you captured more than a few seconds of video. You should be able to get away with selecting 60 or so frames for processing (for instance by selecting only every few frames).
4. The red box in Figure 2 shows the template that was used for registration. You will need to select a similar template from the first frame of your video. You will then search for a match to this template in successive frames. The location of the match will tell you how many pixels your camera has shifted. Choose a size for this template (i.e. 16x16 pixels). You may need to adjust this to improve your results.
5. You only need to search for a template match within a window of your target frame (see Figure 3). The window should be centered on the location of the template in the first frame. The size of the window should be slightly larger than the sum of the template size and the maximum shift of the target object over the entire set of video frames. For instance, in Figure 2 the template was 60x60 pixels and the maximum shift of the template in Figure 2 was about 40 pixels, and a search window of 200x200 pixels was used.

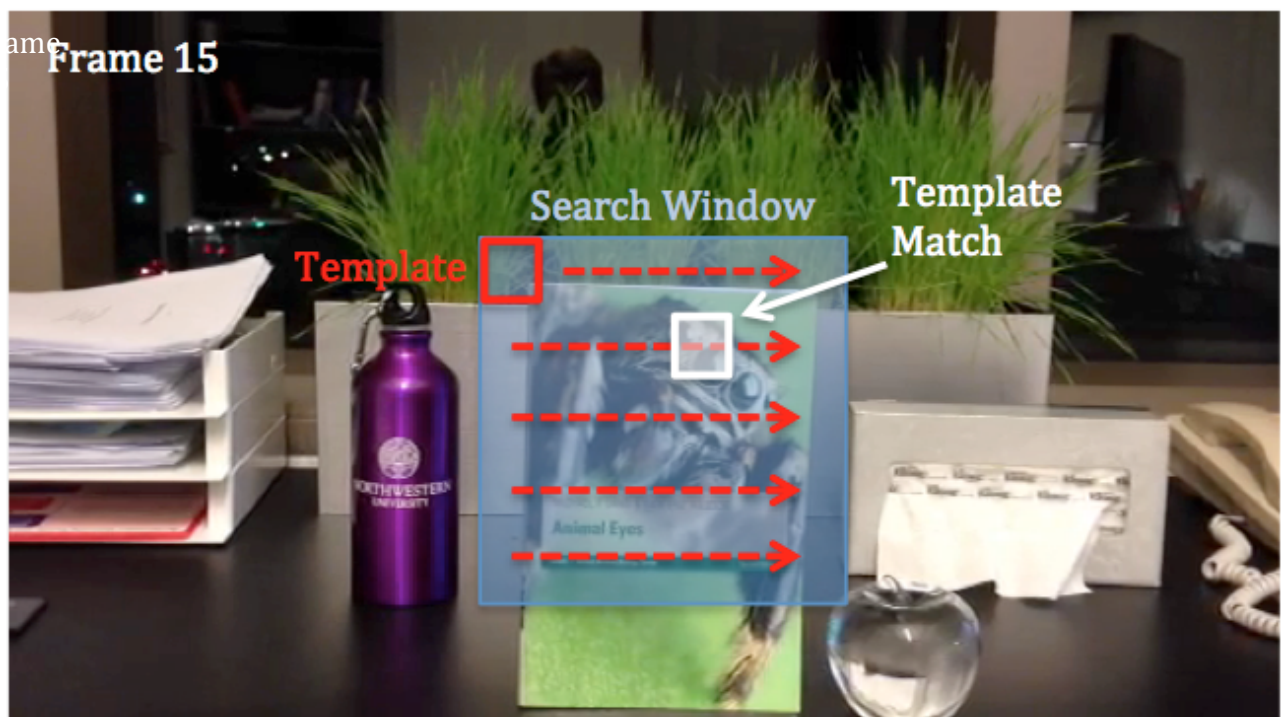


Figure 3: Search within a window of your target frame for a match to your template. The location of this match tells you the amount of shift of the camera relative to the first frame of video.

6. Now, perform an autocorrelation of the template with the extracted search window. If your template $t[n, m]$ is $T \times T$ pixels and your window $w[n, m]$ is $W \times W$ pixels, compute the normalized correlation function $A[i, j]$ using the equation:

$$A[i, j] = \frac{1}{N \cdot P} \sum_{n=1}^T \sum_{m=1}^T t[n, m] \cdot w_{ij}[n, m] \quad (1),$$

where w_{ij} is a block of pixels taken from the window w that is centered on the pixel location (i, j)

$$w_{ij}[n, m] = w^2 \left[n + i - \frac{W}{2}, m + j - \frac{W}{2} \right] \quad (2),$$

and the normalization factors are computed as

$$N = \sum_{n=1}^{T'} \sum_{m=1}^T t^2[n, m] \quad (2)$$

$$P = \sum_{n=1}^{T'} \sum_{m=1}^T w_{ij}^2[n, m] \quad (3).$$

Note that the normalized factor P is computed over a window the size of the template ($T \times T$ pixels) *not over the entire window ($W \times W$ pixels)!*

HINT: Use the builtin Matlab function `nlfilter` to perform the sliding window operation of Equation 1!

The pixel shift $[s_x, s_y]$ for each frame of video can then be computed as:

$$[s_x, s_y] = \underset{i, j}{\operatorname{argmax}} A[i, j] \quad (4).$$

Show a plot of the pixel shift for each frame of video (see Figure 4).

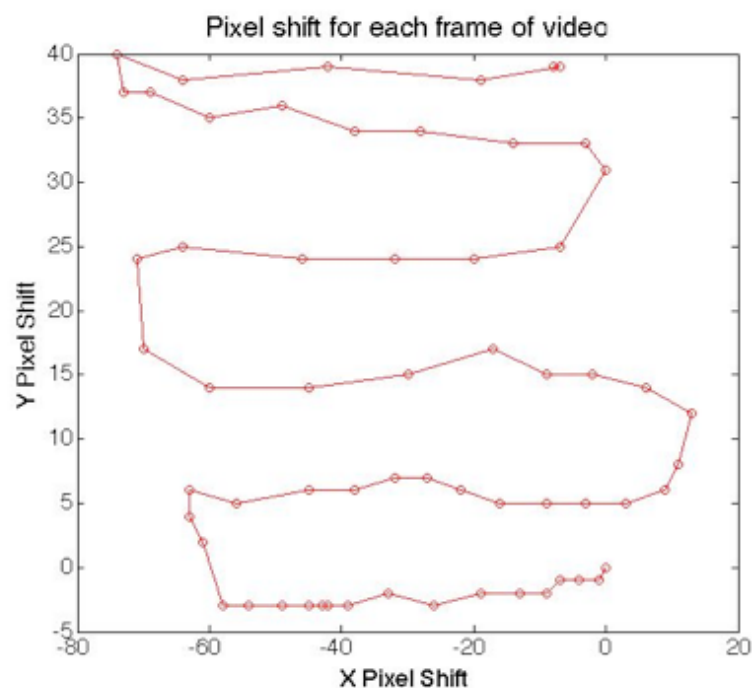


Figure 4: The pixel shifts for the video in Figure 1 found from Equation 4.

3. Create a synthetic aperture photograph (5 points)

Once you have calculated the pixel shift from Equation 4, you can generate a synthetic aperture photograph simply by shifting the images of each frame of video in the opposite direction and then summing up the result. If the $f_i[n, m]$ is the i^{th} video frame, and $[s_x^i, s_y^i]$ is the shift for that frame, you can simply calculate your synthetic aperture photograph as

$$P[n, m] = \sum_{i=1}^N f_i[n - s_x^i, m - s_y^i] \quad (5).$$

You may want to use the builtin Matlab functions *maketform* and *imtransform* to help you do this. Figures 5 and 6 show synthetic aperture photographs generated from the video in Figure 1.

4. Refocus on a new object (2 points)

Now you will refocus on a new object. To do this you will simply select your template from the first frame of video to be centered on a different object. Then all you need to do are repeat steps 3 and 4.



Figure 4: The first frame of video from Figure 1.



Figure 5: A synthetic aperture photograph using a template from the book in front.



Figure 5: A synthetic aperture photograph using a template from northwestern water bottle.

Problems you may run into. You will find that the quality of your synthetic aperture photographs depends a lot on how you moved the camera while capturing the video. This is because our registration algorithm does not take into account possible tilt and rotation of the camera. The more you are able to avoid tilt and rotation, the better the results will be.

What to Submit:

Submit a zip file to the dropbox, which includes

1. A write-up document that explains what you did, what the results are, and what problems you encountered.
2. The video you captured and All Matlab code that you wrote.
3. The write-up should include all the figures that you were instructed to generate.

References

[1] *Synthetic aperture confocal imaging*, Marc Levoy, Billy Chen, Vaibhav Vaish, Mark Horowitz, Ian McDowall, and Mark Bolas. 2004. *ACM Trans. Graph.* 23, 3 (August 2004), 825-834.