EECS 395/495: Introduction to Computational Photography

Homework 6: Synthetic Aperture Imaging

Student Name: Haikun Liu

Student Number: 2903021

NetID: hlg483

Objective: To synthesize an image that appears to be captured from a synthetic aperture that is much larger aperture than the tablet camera

1. Capture an unstructured light field

Using the back camera of the tablet, we can capture video with a zig-zag shooting route to a static scene. The static scene contains some objects with different depths. During we capture the video, we need to avoid tilting and rotating the camera as much as possible. As shown in the Figure 1, we capture an unstructured light field of scene by waving the camera in front of the still scene.

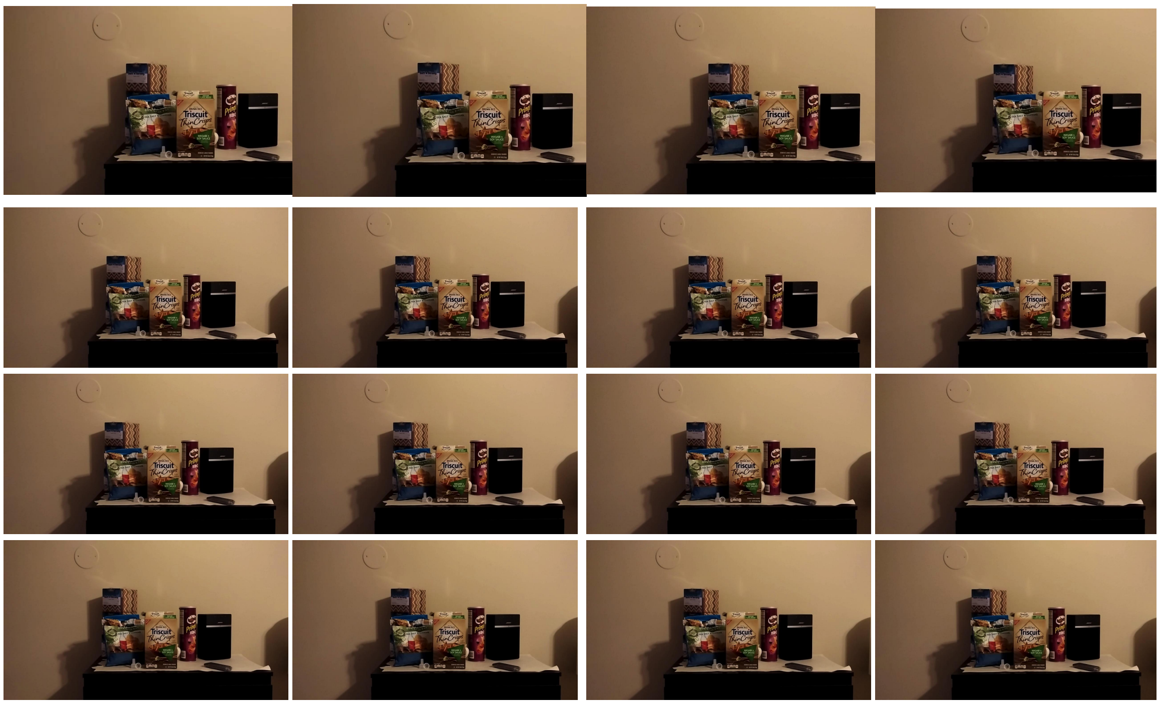

Figure 1: 16 frames from the captured video with the same time interval

2. Register the frames of video using template matching

We begin with loading the video frames into the Matlab and converted them into grayscale. We can take a frame within a certain time interval, say we take one frame from each 100 frames, to speed up subsequent registration process.

```
str_Load = '/Users/HKLHK/Desktop/Northwestern University Q1/EECS495 Introduction to Computational Photography/HW/HW6/VID_20151127_160303.mp4';
v = VideoReader(str_Load);
i=1;
video = read(v);
[row column color numIm] = size(video);
sampled_video=video(:,:,:,1:round(numIm/60):end);
[row column color numIm] = size(sampled_video);
for k = 1: numIm
```

```
    gray_video(:,:,k) = rgb2gray(sampled_video(:,:,:,k));
end
```

Then, we can start to register the frame of video. We can use the following code to select a template from the first frame for matching:

```
[template, rect_template] = imcrop(gray_video(:,:,1));
```

Here, the cropped image within the blue bounded box is chosen as the template. It is shown in the Figure 2.

Figure 2: The blue box shows the template that is used to register the video frames

Next, we calculate the variance image through all of the frames using the code:

```
for i=1:row
    for j=1:column
        var_gray_video(i,j)=sum(double(gray_video(i,j,:)));
    end
end
```

```
normalized_var_gray_video =
uint8(round(var_gray_video/max(max(var_gray_video))*255));
```

Based on the variance image we can roughly choose a search range for each frame to conduct template matching:

```
[search, rect_search] = imcrop(normalized_var_gray_video);
```

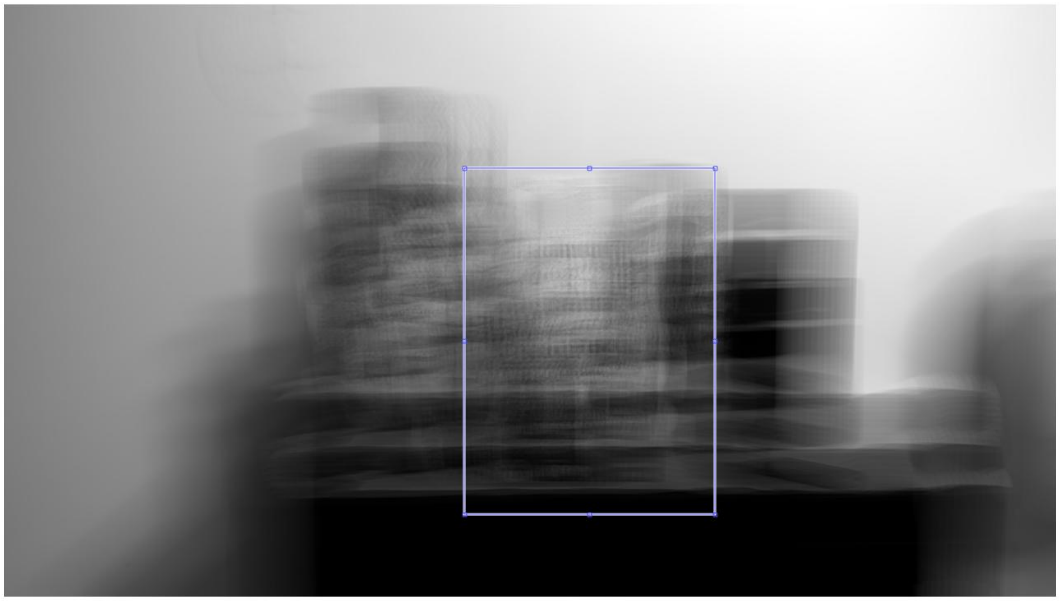

Figure 3: The blue box shows the search window that is used for template matching

The search window is used to crop an area of each frame at the same location with the same size.

```
clear search;

for k = 1: numIm
    search(:,:,k) = imcrop(gray_video(:,:,k),rect_search);
end
```

After we have the template and the search area of each frame, we can conduct template matching of each frame within the search window. The matching process is based on the cross-correlation coefficients between the template and the search position. The position with the largest cross-correlation coefficient is chosen as the pixel shift of each frame.

```
function [ xoffSet,yoffSet ] = correlation( template,searchwindow )
c = normxcorr2(template,searchwindow);
[ypeak, xpeak] = find(c==max(c(:)));
yoffSet = ypeak-size(template,1);
xoffSet = xpeak-size(template,2);
end


XWindowPixelShift = [];
YWindowPixelShift = [];
for k = 1: numIm
   [xoffSet,yoffSet] = correlation( template(:,:),search(:,:,k) );
   XWindowPixelShift(k) = xoffSet;
   YWindowPixelShift(k) = yoffSet;
end

XPixShift=XWindowPixelShift-XWindowPixelShift(1);
YPixShift=YWindowPixelShift-YWindowPixelShift(1);
```

We can plot the pixel shift for each frame using the following code:

```
figure;
scatter(XPixShift,YPixShift,'*');
title('Pixel shift for each frame of video');
ylabel('Y Pixel Shift');
xlabel('X Pixel Shift');
axis('square');
```

The plot is shown in Figure 4:

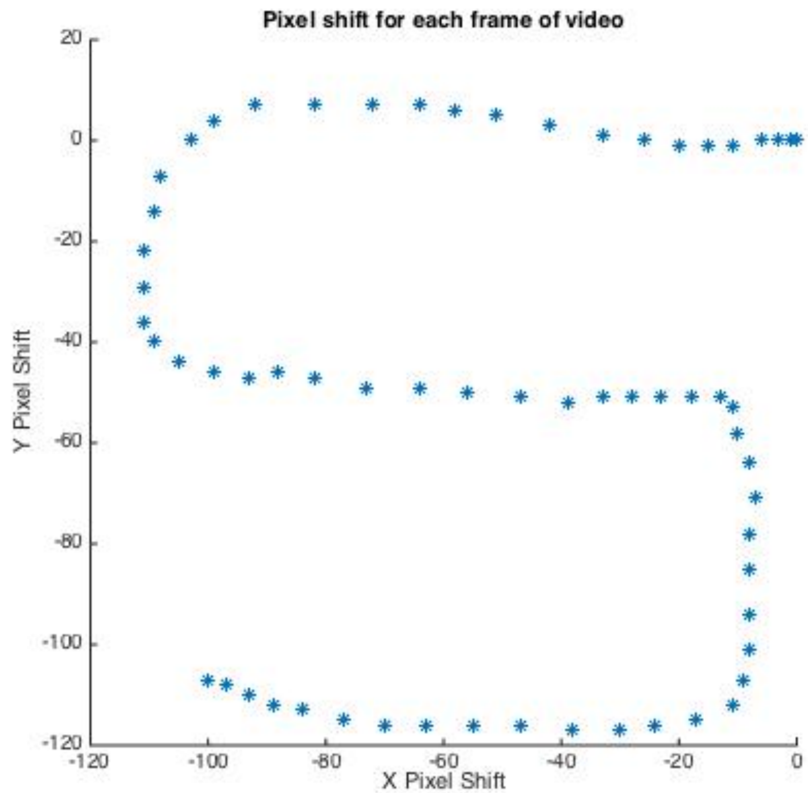

Figure 4: The pixel shifts for the video

3. Create a synthetic aperture photograph

Once we have calculated the pixel shift of each frame in the video, we can generate a synthetic aperture photograph by translating the images of each frame of video in the opposite direction and then taking the average among these shifted frames as the final synthesis result.

```
for k = 1: numIm
    T(:,:,:,k) = imtranslate(sampled_video(:,:,:,k),[-XPixShift(k), -
YPixShift(k)],'FillValues',255);
end

P = uint8(round(sum(T,4)/numIm));
```

```
Crop_P = imcrop(P,[max(abs(YPixShift)) max(abs(XPixShift)) column
row]);
```

The synthetic aperture photograph is shown as Figure 6:

Figure 6: A synthetic aperture photograph using a template from the Triscuit box in front.

4. Refocus on a new object

Now we can refocus on a new object. This time we can choose the icon of Pringles chip box as the template.

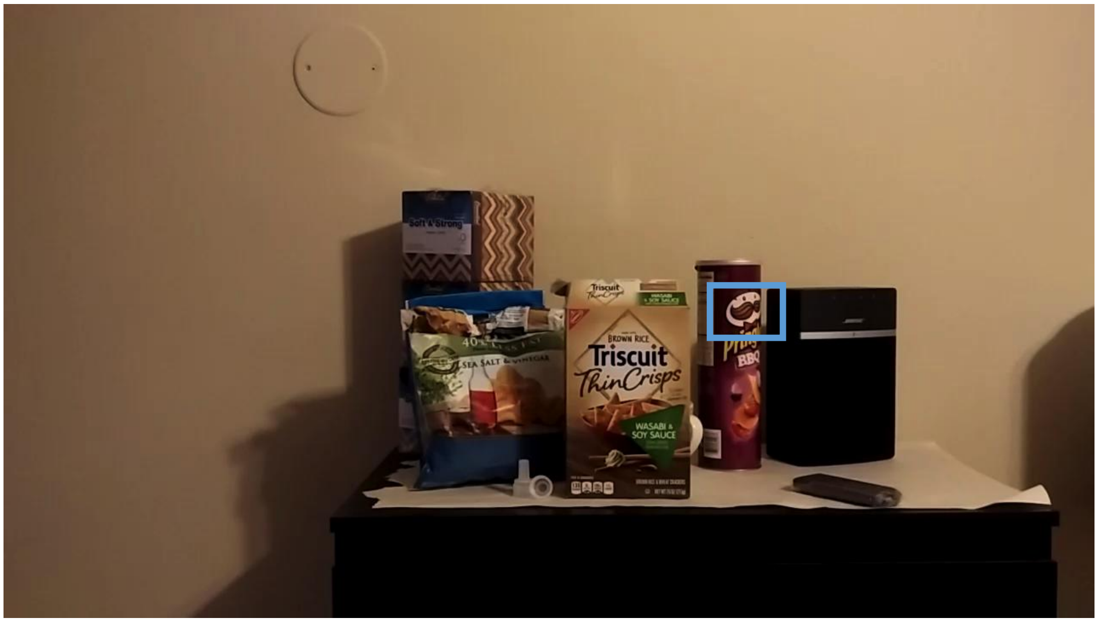

Figure 7: The first frame of video

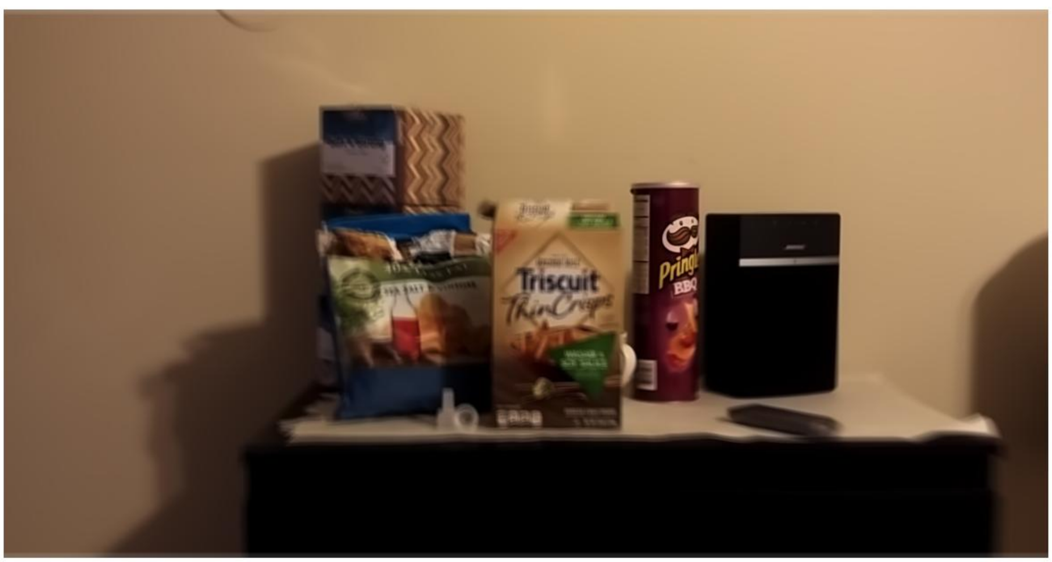

Figure 8: A synthetic aperture photograph using a template from the icon of Pringles chip box