Homework 3 Questions

Instructions

- 4 questions.
- Write code where appropriate.
- Feel free to include images or equations.
- Please make this document anonymous.
- **Please use only the space provided and keep the page breaks.** Please do not make new pages, nor remove pages. The document is a template to help grading.
- If you really need extra space, please use new pages at the end of the document and refer us to it in your answers.

Questions

Q1: Imagine we were tasked with designing a feature point which could match all of the following three pairs of images. Which real world phenomena and camera effects might cause us problems? Use the MATLAB function *corner* to investigate. $corner(I, 1000)$.

RISHLibrary — Chase — LaddObservatory

A1: If the picture is blurred by the camera, feature point detection does not work well. When I execute "corner(I, 1000)" about chase1.jpg and chase2.jpg(blurred version), I can get 1000 feature points in the chase1.jpg but I can get just 784 points in the chase2.jpg.

In the observatory1.jpg vs observatory2.jpg case, there are many disturbance factor (bush, human, etc) about observing feature points of observatory in the observatory2.jpg than observatory1.jpg. So as you can see in the Q1 appendix, in the result of executing "corner(I, 1000)", less meaningful feature points are detected more when disturbance factor exist by real world phenomena.

RISHLibrary.jpg case is similar, both two picture have disturbance factor about observing feature points of the pattern of the floor. Futhermore, in the RISHLibrary2.jpg case, there is exceptionally bright dark part, so there is a tendency for the detected feature point to move to that part. Therefore, that facts cause difficult to find and match all the same feature points in image pairs.

Q2: In designing our feature point, what characteristics might we wish it to have? Describe the fundamental trade-off between feature point invariance and discriminative power. How should we design for this trade-off?

A2: Your answer here.

We wish detected feature points are repeatable and distinctive, in other words, feature points are much better when they have high invarianve and high discriminative power. But it is difficult to satisfy both condition at the same time. Furthermore, this trade-off must vary from task to task and no single method can be optimal in all situations. So finding trade-off between discriminative power and invariance is essential, for example, using covariant detection and invariant description.

Q3: In the Harris corner detector, what do the eigenvalues of the 'M' second moment matrix represent? Discuss both how they relate to image intensity and how we can interpret them geometrically.

A3: Your answer here.

$$E(u,v)=\sum_{x,y}\underbrace{w(x,y)}_{\text{Window function}}\Big[\underbrace{I(x+u,y+v)}_{\text{Shifted intensity}}-\underbrace{I(x,y)}_{\text{Intensity}}\Big]^2$$

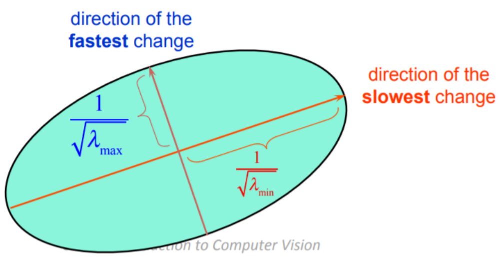
In the above formula, E(u,v) indicate change of intensity for the shift [u,v]. For small

$$E(u,v) \approx [u \;\; v] \; M \begin{bmatrix} u \\ v \end{bmatrix} \qquad M=\sum_{x,y} w(x,y)\begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix}$$

shifts [u,v], we can have a quadratic approximation like above. second moment matrix M is as follows.

M is symmetric marix, so sum of eigenvalues of M is same to trace of M and product of eigenvalues of M is same to det(M). In other words, sum and product of eigenvalues of M are indicated by expression about change of image intensity with x direction and y direction.

If we consider a horizontal slice of E(u, v), that is become to equation of an ellipse and lengths of axis are determined by the eigenvalues, like the below figure.



Therefore, in the view geometrically, when eigenvalues are larger, E(u,v) changes rapidly. So the probability that point is a special feature point increases.

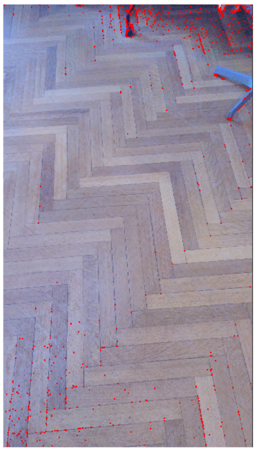
Q4: Explain the difference between the Euclidean distance and the cosine similarity metrics between descriptors. What might their geometric interpretations reveal about when each should be used? Given a distance metric, what is a good method for feature descriptor matching and why?

A4: Your answer here.

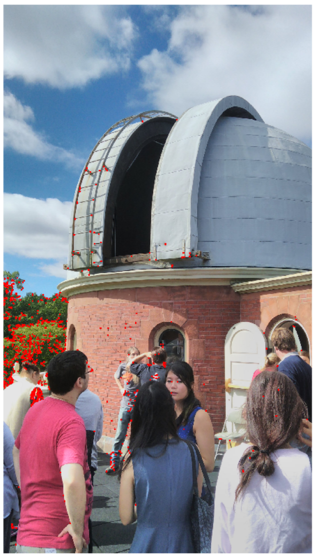
The biggest difference between the Euclidean distance and the cosine similarity metrics is that point, magnitude of the vectors are ignored when use cosine similarity metrics. So when both directions and magnitude of vectors pair need to be considered it is better to use Euclidean distance. When just direction of vector pair need to be considered, it is better to use cosine similarity metrics, for example, when to discard the magnitude of a vector because magnitude of the vector become larger without meaning anything different. In the feature descriptor matching, usually the magnitude of two vectors are needed, for example, magnitude of gradiant, etc. So Euclidean distance is more useful than cosine similarity metrics.

Q1 - appendix (red points are corner)

left : chase1, right : chase2



left : RISHLibrary1, right : RISHLibrary2



up : LaddObservatory1, down : LaddObservatory2

```
A = imread('chase1.jpg');
B = imread('chase2.jpg');

C1 = corner(rgb2gray(A),1000);
C2 = corner(rgb2gray(B),1000);

[l,c] = size(A);
for i=1:length(C1)
    x = C1(i,2);
    y = C1(i,1);
    for j=x-5:x+5
        for k=y-5:y+5
            if j<1 || k<1 || k>c || j>l
                continue;
            end
            A(j,k,:) = [255,0,0];
        end
    end
end
for i=1:length(C2)
    x = C2(i,2);
    y = C2(i,1);
    for j=x-5:x+5
        for k=y-5:y+5
            if j<1 || k<1 || k>c || j>l
                continue;
            end
            B(j,k,:) = [255,0,0];
        end
    end
end

imshow(A);
imshow(B);
```

using this code to draw corner upon picture.