Final Review
CMSC 733
Fall 2013

We have covered a lot of material in this course. One way to organize this material is around a set of key equations and algorithms. You should be familiar with all of these, and able to apply them to problems. I do not want you to memorize these equations, but to understand them and be able to use them. I am also listing some sample problems. It's a bit hard to come up with lots of good questions, so some of these may be easier or harder than questions I would ask on the final. Also, I came up with a lot of questions quickly, so if one of these questions seems ill-considered, it probably is, and you shouldn't worry too much about it.

This document is intended to give your studying for the final some focus. However, I do not promise that every question on the final will relate to something that is mentioned in this review.

1. Convolution and Fourier Transforms

You should be familiar with the Fourier series, and understand that any function can be expressed as an infinite sum of harmonic terms:

$$f(t) = a_0 + \sum_{1}^{\infty} a_k \cos(kt) + b_k \sin(kt)$$

You should also understand the Fourier transform, and understand that the coefficients of a Fourier transform or series can be obtained by taking an inner product, eg.:

$$F(k) = \int_{-\infty}^{\infty} f(t) e^{-ikt} dt$$

You should also understand the meaning of convolution, and be able to determine the effects of convolution. In 1D this is:

$$h(t) = \int_{-\infty}^{\infty} f(t-u) g(u) du$$

And you should understand the meaning and significance of the convolution theorem.

You should be able to answer questions such as:

a. What is the result of convolving a 1D image: [1, 2, 1, 3, 4, 5, 1] with a filter: [-1 2 -1].
b. What is the result of convolving cos(2t) with cos(4t)?
c. Compute the first three terms of the Fourier series representation of the function f, where f(x) = 1 for $-1<x<1$, f(x) = 0 otherwise.

2. Linear and non-linear diffusion. You should understand the 2D and 1D diffusion equations, such as, in 2D:

$$\frac{\partial u}{\partial t} = \mathrm{div}(D\nabla u)$$

along with the equations for flux and conservation:

$$j = -D\nabla u \qquad \frac{\partial u}{\partial t} = -\mathrm{div} j.$$

You should also understand how isotropic, linear diffusion can be solved using convolution, and understand diffusion as a Markov process. You should understand the idea of nonlinear diffusion, and how different choices of D give you anisotropic and/or nonlinear diffusion. Some sample problems might include:

a. Suppose we have an image in which the intensity is described by I(x,y) = 7 + x + xy. If D = [1;0;0;1] what is the flux at (7,3)? What is the change in intensity with respect to time at (7,3)?
b. Given that in Perona-Malik diffusion, D is defined using:

$$g(\|\nabla u\|^2) = \frac{1}{\sqrt{1 + \frac{\|\nabla u\|^2}{\lambda^2}}}$$

if lambda = 1, what is the Perona-Malik diffusion at location (x,y)?

3. Edge Detection. Understand Canny edge detection as an algorithm with the following steps; 1) Smooth the image; 2) Compute the gradient; 3) Find local gradient extrema that are a) above some threshold; and b) bigger than their neighbors in the gradient direction.

a. Suppose we have smoothed an image, and a 3x3 region in the image has intensities of:

3	5	12	13	14
4	6	13	14	15
5	7	14x	15	16
6	8	15	16	17
7	9	16	17	18

How would you decide whether the pixel in the middle, marked by an 'x', is an edge?

4. Markov Processes, Markov Random Fields. You should understand what makes something Markov, what makes something a Markov Random Field, the relationship between an MRF and a Gibbs distribution, how to determine the steady state of a Markov process, and how MRFs can be solved using graph cuts. You should recognize the equations:

$$P(f_i | f_{S-\{i\}}) = P(f_i | f_{N_i})$$

$$P(f) = \frac{1}{Z} e^{-\frac{U(f)}{T}} \qquad U(f) = \Sigma_{c \in C} V_c(f)$$

Some sample problems:

a. Suppose Aaron and Betty play the following game. Each starts with \$50. At every turn, they flip a biased coin, that is heads with probability .6. If the coin is heads, Aaron gives Betty a dollar. If it is tails, Betty gives Aaron a dollar. If either player ever goes bankrupt, the other player gives them \$20. If the game is played for a long time, what is the expected amount of money that Betty will have? (This might be hard to do with pencil and paper, so just figure out how you'd solve it using Matlab, ideally using eigenvectors).

b. Suppose every pixel of an image is to be labeled boat, sky, or water. Every pixel has a 90% chance of having the same label as its immediate neighbor, and a 5% chance of having each different label. If a pixel is blue, it has a 50% chance of being water and a 50% chance of being sky. If it is not blue, it has a uniform chance of having any label. Explain how you would create an MRF to model this problem.

5. Normalized Cut and bilateral filtering. You should remember what bilateral filtering is, and understand the equations:

$$\mathbf{h}(\mathbf{x}) = k_d^{-1}(\mathbf{x}) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathbf{f}(\xi) c(\xi, \mathbf{x}) \, d\xi$$

$$c(\xi, \mathbf{x}) = e^{-\frac{1}{2}\left(\frac{d(\xi,\mathbf{x})}{\sigma_d}\right)^2}$$

where

$$d(\xi, \mathbf{x}) = d(\xi - \mathbf{x}) = ||\xi - \mathbf{x}||$$

is the Euclidean distance between ξ and $\mathbf{x}$. The similarity function s is perfectly analogous to c:

$$s(\xi, \mathbf{x}) = e^{-\frac{1}{2}\left(\frac{\delta(\mathbf{f}(\xi),\mathbf{f}(\mathbf{x}))}{\sigma_r}\right)^2}$$

where

$$\delta(\phi, \mathbf{f}) = \delta(\phi - \mathbf{f}) = ||\phi - \mathbf{f}||$$

For normalized cut, you should know what a graph cut is, and understand the normalized cut cost function:

Min Ncut(A,B) = cut(A,B)/assoc(A,V) + cut(B,A)/assoc(B,V)

You don't need to remember the entire derivation, but you should understand that a relaxed version of normalized cut is solved by solving:

$$\frac{z^T D^{-\frac{1}{2}}(D-W)D^{-\frac{1}{2}} z}{z^T z}$$

and remember the steps of the normalized cut algorithm needed to formulate and solve this problem. Some sample problems might be:

a. Compare the results of bilateral filtering and Perona-Malik diffusion on a step edge.
b. Suppose we have an image containing a white disk on a black background. Using normalized cut, as described by Shi and Malik, what are the possible results we can get? How do they depend on the size of the disk, the size of the image, and the parameters of normalized cut?

6. E-M and K-means. You should know the steps of both algorithms. You should understand that K-means is minimizing:

$$\sum_{i=1}^{n} (m - x_i)^2$$

You should understand what a Gaussian mixture model is, as described by:

$$p(x; m, \sigma) = \sum_{k=1}^{K} p_k g(x; m_k, \sigma_k)$$

and you should understand the E

$$p^{(i)}(k \mid n) = \frac{p_k^{(i)}\, g(\mathbf{x}_n;\, \mathbf{m}_k^{(i)}, \sigma_k^{(i)})}{\sum_{m=1}^{K} p_k^{(i)}\, g(\mathbf{x}_n;\, \mathbf{m}_k^{(i)}, \sigma_k^{(i)})}$$

and M steps:

$$\mathbf{m}_k^{(i+1)} = \frac{\sum_{n=1}^{N} p^{(i)}(k \mid n)\, \mathbf{x}_n}{\sum_{n=1}^{N} p^{(i)}(k \mid n)}$$

$$\sigma_k^{(i+1)} = \sqrt{\frac{1}{D} \frac{\sum_{n=1}^{N} p^{(i)}(k \mid n)\, \|\mathbf{x}_n - \mathbf{m}_k^{(i+1)}\|^2}{\sum_{n=1}^{N} p^{(i)}(k \mid n)}}$$

$$p_k^{(i+1)} = \frac{1}{N} \sum_{n=1}^{N} p^{(i)}(k \mid n)\,.$$

a. Give an example of three points for which K-means will converge to different answers, depending on how it is initialized.
b. Give an example in which K-means will converge to a local minima that is not a global minima, but for which E-M will only converge to the global minimum.

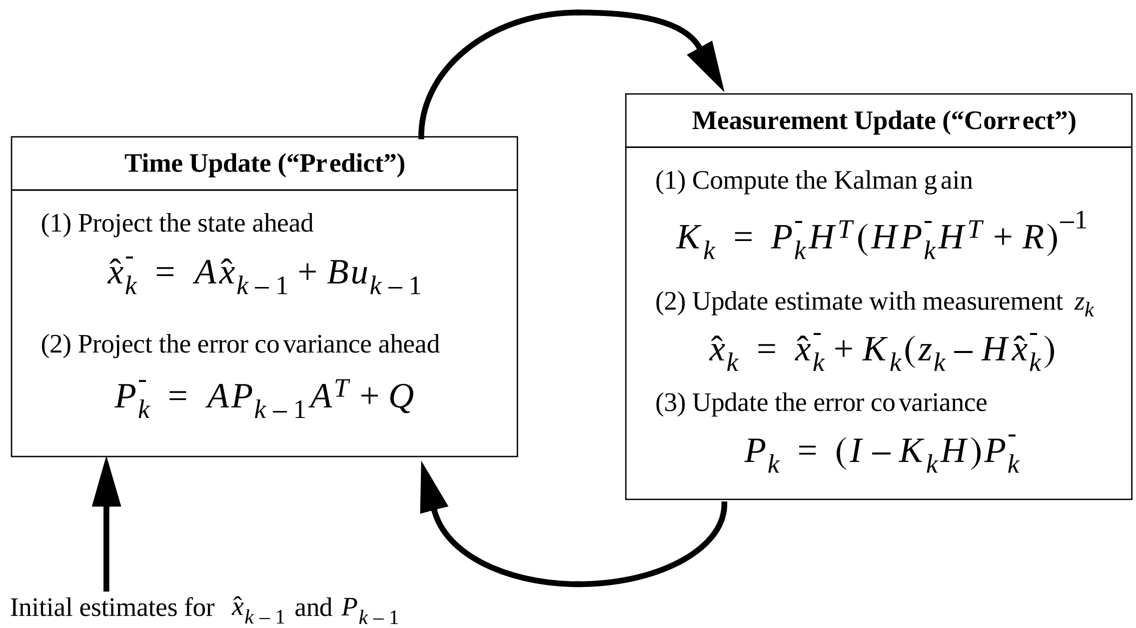
7. Background subtraction and Texture. We talked about how background subtraction and texture analysis can be done by modeling these as statistical processes. You should understand how a Gaussian mixture model could be used for background subtraction, and how texture can be modeled using textons (and what the relationship is between textons and bag-of-words

models) and as a Markov process. (Questions on these topics are likely to be really questions about statistical modeling.)

8. SIFT descriptors and blob detectors. Know what a SIFT descriptor and blob detector are and how they work.

    a. How would the SIFT descriptor for a region of an image change if the lighting got brighter so that every pixel became twice as bright?

9. Tracking. You should understand the basic outline of the Kalman filter and particle filter, as shown in these diagrams:



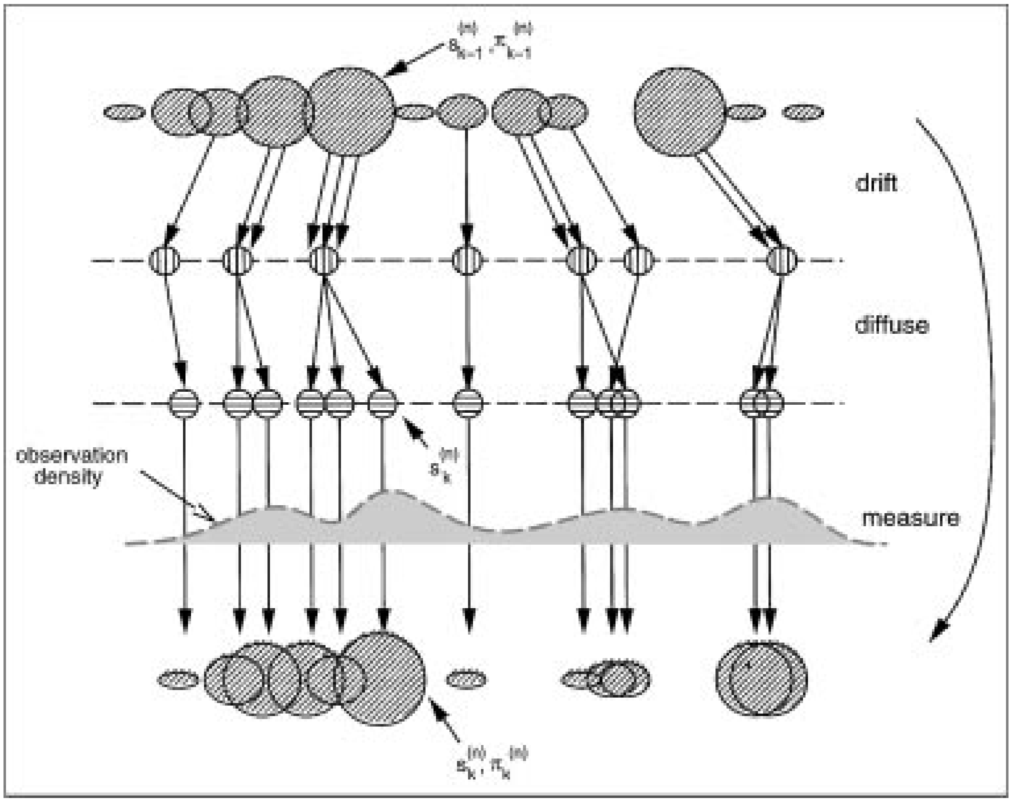


Figure 5. One time-step in the CONDENSATION algorithm: Each of the three steps—drift-diffuse-measure—of the probabilistic propagation process of Fig. 2 is represented by steps in the CONDENSATION algorithm.

a. Give an example of a problem for which the particle filter could perform tracking but the Kalman filter could not. Is there any problem for which Kalman filtering would work, but particle filters would not?
b. How would you construct a Kalman filter to track an object that moves in circles (with perhaps a little noise in its motion)?

10. Perspective projection and projective transformations. You should understand how perspective projection works. You should know what the horizon is and what a vanishing point is. You should know how to represent the relationship between two images using projective transformations, and what homogenous coordinates are.

$$(x,y,z) \rightarrow (f\frac{x}{z}, f\frac{y}{z})$$

a. Suppose we have a camera with a focal point at the origin and an image plane of z = 1. Give an example of parallel lines in the world that have a vanishing point at (3,2,1) in the image plane.

b. Suppose we have four 2D points. Under what circumstances does there exist a projective transformation that will map these to the locations (0,0), (1,0), (0,1), (1,1)?

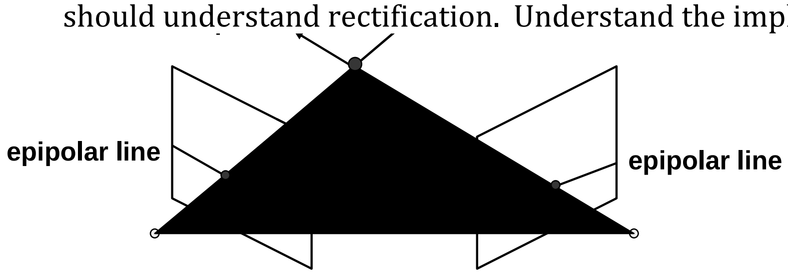
c. Prove that a circle in the world will always project to an ellipse in the image.

11. Stereo matching. You should be familiar with using correlation, dynamic programming, or graph cuts (alpha-beta swap and alpha expansion) for stereo matching.

    a. Give an example of a scene for which you would expect graph cuts to work better than dynamic programming in stereo matching.

    b. Give an example in which dynamic programming would outperform graph cuts.

12. Epipolar geometry and triangulation. You should understand how the epipolar constraint is derived, and what the epipole is. You should understand how to use triangulation to find the depth of an object. And you should understand rectification. Understand the implications of this picture:



and this equation:

$$Z = f\frac{T}{d}$$

    a. Suppose we take a pair of stereo images, but one camera is on top of the other (for example, focal points at (0,0,0) and (0,10,0), and image plane at z = 1. What are the epipolar lines?

    b. Pick two image points that could be matched consistently with this epipolar geometry. Determine their depth.

13. The eight point algorithm. Understand what the essential matrix is. Why is it rank deficient? How does it encode epipolar constraints?

$$\hat{x}_1^T E \hat{x}_0 = 0, \quad E = [t]_\times R \quad \hat{x}_1^T([t]_\times R)\hat{x}_0 = 0.$$

    a. Consider our standard stereo setup, with focal points at (0,0,0) and (10,0,0) and an image plane of z = 1. What is the essential matrix for this setup? (To answer this, you have to have a way of turning points on the image plane into 2D points. For example, when the focal point is at (10,0,0) we could consider the point (10,0,1) to be the origin of the second image).

14. Optical flow and corner detection. Understand the optical flow equation and how it is derived

$$0 = I_t + \nabla I \cdot [u \;\; v]$$

and the equation for the Lucas-Kanade solution to optical flow.

$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}$$

Also understand how to use the matrix on the left to find corners.

   a. Make up an image, and find the optical flow at different points on it.
   b. Implement a corner detector based on the above and try it on some images (this is only a few lines of code).

15. Factorization for structure-from-motion. You should understand how to represent scaled orthographic projection as matrix multiplication. You should understand the rank theorem.

Scale — Projection — 3D Translation — 3D Rotation

$$s\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \end{pmatrix} \begin{pmatrix} r_{1,1} & r_{1,2} & r_{1,3} & 0 \\ r_{2,1} & r_{2,2} & r_{2,3} & 0 \\ r_{3,1} & r_{3,2} & r_{3,3} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} P$$

$$\equiv \begin{pmatrix} s_{1,1} & s_{1,2} & s_{1,3} & st_x \\ s_{2,1} & s_{2,2} & s_{2,3} & st_y \end{pmatrix} P$$

$$\begin{pmatrix} \tilde{u}_1^1 & \tilde{u}_2^1 & . & . & . & \tilde{u}_n^1 \\ \tilde{v}_1^1 & \tilde{v}_2^1 & & & & \tilde{v}_n^1 \\ \tilde{u}_1^2 & \tilde{u}_2^2 & & & & \tilde{u}_n^2 \\ \tilde{v}_1^2 & \tilde{v}_2^2 & & & & \tilde{v}_n^2 \\ . & . & & & & . \\ . & . & & & & . \\ . & . & & & & . \\ \tilde{u}_1^m & \tilde{u}_2^m & & & & \tilde{u}_n^m \\ \tilde{v}_1^m & \tilde{v}_2^m & . & . & . & \tilde{v}_n^m \end{pmatrix} = \begin{pmatrix} s_{1,1}^1 & s_{1,2}^1 & s_{1,3}^1 \\ s_{2,1}^1 & s_{2,2}^1 & s_{2,3}^1 \\ s_{1,1}^2 & s_{1,2}^2 & s_{1,3}^2 \\ s_{2,1}^2 & s_{2,2}^2 & s_{2,3}^2 \\ & . & \\ & . & \\ & . & \\ s_{1,1}^m & s_{1,2}^m & s_{1,3}^m \\ s_{2,1}^m & s_{2,2}^m & s_{2,3}^m \end{pmatrix} \begin{pmatrix} x_1 & x_2 & . & . & . & x_n \\ y_1 & y_2 & & & & y_n \\ z_1 & z_2 & & & & z_n \end{pmatrix}$$

a. If we assume scaled orthographic projection, show that a set of points can be reconstructed up to an affine transformation from two images.
b. Show that the structure of the points up to a similarity transformation cannot be determined uniquely from two images.

16. Bag-of-words algorithm for classification. Understand the basic idea of bag of words classification.
17. Illumination. Understand lambertian reflectance. Understand how the set of images of a lambertian object form a linear subspace in the absence of shadows.
    a. Suppose we take a set of images of a Lambertian object as the sun moves in a circle around the earth. Ignore all shadows, and assume all light comes directly from the sun. If we form a matrix of the images we get, with each row holding a different image, and each column corresponding to a pixel, what will be the rank of that matrix?