ASSIGNMENT #5 (EC720 A1)
"Digital Video Processing"
Date: November 14, 2017
Due date: November 29, 2017

1. *Image sequence spectrum* (25 points).
This problem extends the spectral properties of constant-velocity motion, that we considered in class, to constant-acceleration motion. Suppose a continuous moving image $s_c(\boldsymbol{x}, t)$, $\boldsymbol{x} = [x\ y]^T \in R^2$, $t \in R$, is a function of velocity $\boldsymbol{v} = [v_1\ v_2]^T$ and acceleration $\boldsymbol{a} = [a_1\ a_2]^T$ (uniform global motion under acceleration):

$$s_c(\boldsymbol{x}, t) = s_c(\boldsymbol{x} - \boldsymbol{v}t - \frac{1}{2}\boldsymbol{a}t^2, 0) \triangleq s_0(\boldsymbol{x} - \boldsymbol{v}t - \frac{1}{2}\boldsymbol{a}t^2),$$

where $s_0(\boldsymbol{x}), \boldsymbol{x} \in R^2$ is a continuous still image.

   (a) Exploiting the properties of Dirac impulse δ show that the Fourier transform of the signal $s_c(\boldsymbol{x}, t)$ can be expressed as follows:

   $$\mathcal{F}\{s_c(\boldsymbol{x}, t)\} = S_0(\boldsymbol{f})[\delta(\boldsymbol{f}^T\boldsymbol{v} + f_t) \overset{f_t}{*} Q(\boldsymbol{f}, f_t)]$$

   where $S_0(\boldsymbol{f}) = \mathcal{F}_{\boldsymbol{x}}\{s_0(\boldsymbol{x})\}$, $Q(\boldsymbol{f}, f_t) = \mathcal{F}_t\{e^{-j\pi\boldsymbol{f}^T\boldsymbol{a}t^2}\}$, $\boldsymbol{f} = (f_x\ f_y)^T$ is the spatial frequency, f_t is the temporal frequency, and $\overset{f_t}{*}$ is a convolution with respect to f_t. Note that $\mathcal{F}_{\boldsymbol{x}}$ is the 2-D Fourier transform with respect to $\boldsymbol{x}$, while $\mathcal{F}_t$ is the 1-D Fourier transform with respect to t.

   (b) Show that for the special case of no acceleration ($\boldsymbol{a} = [0\ 0]^T$), we obtain:

   $$\mathcal{F}\{s_c(\boldsymbol{x}, t)\} = S_0(\boldsymbol{f})\delta(\boldsymbol{f}^T\boldsymbol{v} + f_t).$$

   (c) Derive an expression for $Q(\boldsymbol{f}, f_t)$ (*Hint:* $\int_{-\infty}^{\infty} e^{js^2} ds = \sqrt{\pi}e^{j\pi/4}$). What impact does the acceleration $\boldsymbol{a}$ have on $\mathcal{F}\{s_c(\boldsymbol{x}, t)\}$? What are the consequences of this impact for the estimation of motion in the frequency domain?

2. *Deinterlacing filters* (15 points)
For each of the following deinterlacing filters:

   (a) vertical two-line averaging: $h_v = [1, 2, 1]^T/2$,
   (b) vertical four-line averaging: $h_v = [1, 0, 7, 16, 7, 0, 1]^T/16$,
   (c) temporal zero-hold (field merging): $h_t = [0, 1, 1]$,
   (d) temporal two-field averaging: $h_t = [1, 2, 1]/2$,
   (e) joint non-causal line and field averaging: $h_{vt} = [0, 1, 0; 1, 4, 1; 0, 1, 0]/4$,
   (f) joint causal line and field averaging:
   $h_{vt} = [0, -1, 0; 0, 0, 0; 0, 9, 0; 0, 32, 16; 0, 9, 0; 0, 0, 0; 0, -1, 0]/32$,

where the rows of the 2-D impulse response matrices h correspond to the temporal direction, whereas the columns correspond to the vertical direction,

(a) derive analytically each filter's magnitude and phase response,

(b) identify if the filter is low-pass, band-pass or high-pass, and zero-phase, linear-phase, or non-linear phase; justify your answer,

(c) plot each filter's impulse response (use `stem`), and compute its 2-D magnitude response and plot it using both `mesh` and `contour` plots,

(d) comment on the shape of the pass band as compared to the ideal lowpass deinterlacing filter (diamond-shaped).

With your analytical solutions, submit the plots in 3-by-3 layout per page (one filter per row: impulse response, magnitude response as mesh, magnitude response as contour).

3. *Deinterlacing – Matlab* (30 points)
Load *RGB* interlaced sequences *horses_interl* and *birds_interl* available from the course web site into a 3-D array (x-y-t). Each sequence consists of 60 TIFF files, each with a 640×180 field. Use the `movie` function to display the sequence in motion (you need to first display with `imshow` and then capture the image into a movie frame with `getframe`).

(a) A proper interpretation of interlaced sequences requires the knowledge of the vertical offset of the first field (either 0 or 1/2 inter-line distance). This is also known as the "first-field top" or "first-field bottom" mode. Before you continue with the experiments, identify the vertical offset of the first field by composing the first frame from the first two fields in both "first-field top" and "first-field bottom" modes, and by visually inspecting the result. In which mode is each sequence?

(b) Apply each of the 6 interpolators from Problem 2 to the two interlaced test sequences. For computational efficiency and to simplify programming **you should use** the `filter2` function in the $y-t$ plane. Note that allocating space to an array prior to processing significantly speeds up the computations. Function `squeeze`, that removes singleton dimensions, is helpful when manipulating video in $y-t$ plane.

(c) Compute the mean-squared error between each of the 6 deinterlaced sequences and the corresponding progressive sequence available on the course web site (*horses_prog* and *birds_prog*). With your solutions, include a table with all mean-squared errors.

(d) Display each deinterlaced sequence in motion on your screen and inspect the result for distortions, resolution loss, etc. Comment on the results obtained. For each filter explain how its pass band affects the deinterlacing quality. Try to pick the filter that performs best for each sequence and explain your choice.

(e) In addition to viewing a sequence, it is helpful to inspect individual deinterlaced frames. With your report include one deinterlaced frame for: four-line averaging, two-field averaging and joint non-causal line and field averaging. Print one frame per page (maximizing frame's size) for the total of 6 pages.

4. *Video compression – Matlab* (30 points)
Motion plays a pivotal role in video compression. Follow the steps below to investigate.

(a) Read the first 10 frames from progressive sequences *horses_prog* and *birds_prog* (640 pixels/line, 360 lines/frame) and compute luminance from each (`rgb2gray`).

(b) *Intra mode*: Assume that each sequence is going to be compressed in *intra mode*, i.e., with no reference to past frames. For each sequence, calculate the histogram of frame #2, and entropy of all frames #2-#10 together (one single entropy of all frames, not a separate entropy for each frame).

Warning: Matlab's `entropy` function assumes an `uint8` array (0-255) as input. However, in both *inter modes* below, a temporal prediction error is computed that has values between -255 and +255. If this error has almost all values between -128 and +128, then you can use `entropy` but you must add an offset to make it non-negative. For an accurate entropy, you should write your own entropy function.

(c) *Inter mode 1*: Apply first-order temporal prediction with coefficient $\alpha[1] = 1.0$ to both sequences, resulting in 9 prediction error (difference) frames. Calculate a histogram of the prediction error for frame #2 and entropy of the prediction error for *all* frames #2-#10 together. Repeat these calculations for $\alpha[1] = 0.9$ but round the predicted value to an integer, i.e., `nint(alpha*image(i,j))`.

(d) *Inter mode 2*: Apply first-order motion-compensated (MC) temporal prediction with coefficient $\alpha[1] = 1.0$ to both sequences using block-matching motion estimation with the absolute-value error metric, full-pixel precision, $N \times N$ blocks and search range $[-D_{max}, D_{max}]$. Use the block matching code you developed earlier in Assignment #3. Using the estimated motion vectors, calculate motion-compensated prediction error for both sequences. Then, calculate the histogram of motion-compensated prediction error for frame #2 as well as entropy of the prediction error for all frames #2-#10 together. Repeat the above experiment for N= 32, 16, 8.

Print your results in 3×2 array as follows (4 pages for each sequence):

- Fig. 1, (3,2,1): frame #2 of the sequence
- Fig. 1, (3,2,2): histogram of frame #2 (entropy of all frames #2-#10 in the title)
- Fig. 1, (3,2,3): temporal prediction error for frame #2 for $\alpha[1] = 0.9$
- Fig. 1, (3,2,4): histogram of this error (entropy of all prediction errors in the title)
- Fig. 1, (3,2,5): temporal prediction error for frame #2 for $\alpha[1] = 1.0$
- Fig. 1, (3,2,6): histogram of this error (entropy of all prediction errors in the title)

- Fig. 2, (3,2,1): MC prediction error for frame #2 for $\alpha[1] = 1.0$, $N = 8$, $D_{max} = 1$,
- Fig. 2, (3,2,2): histogram of this error (entropy of all prediction errors in the title)
- Fig. 2, (3,2,3): MC prediction error for frame #2 for $\alpha[1] = 1.0$, $N = 8$, $D_{max} = 5$
- Fig. 2, (3,2,4): histogram of this error (entropy of all prediction errors in the title)
- Fig. 2, (3,2,5): MC prediction error for frame #2 for $\alpha[1] = 1.0$, $N = 8$, $D_{max} = 10$
- Fig. 2, (3,2,6): histogram of this error (entropy of all prediction errors in the title)

- Figs. 3 and 4: the same as Fig. 2 but for $N = 16$ and $N = 32$.

Draw conclusions from these experiments by answering the following questions. Can inter-frame coding help achieve higher compression? Does the search range D_{max} affect achievable compression ratio? Does the block size have impact on achievable compression ratio? Is there any drawback in using smaller block sizes?