ASSIGNMENT #2 (EC720 A1)
"Digital Video Processing"
Date: September 27, 2017
Due date: October 11, 2017

1. *Motion detection* (50 points).

   Consider motion detection at location $\boldsymbol{x}_0$ based on the measurement of temporal intensity differences $\psi_k[\boldsymbol{x}_0] = I_k[\boldsymbol{x}_0] - I_{k-1}[\boldsymbol{x}_0]$, where I_k denotes an image captured at time t_k. Let Ψ_k be a random field associated with measurements ψ_k and let E_k be a binary random field of motion labels (0 or $\mathcal{S}$ for stationary and 1 or $\mathcal{M}$ for moving) whose realization is denoted e_k. Furthermore, let realizations $e_k^{\mathcal{S}}[\boldsymbol{x}]$ and $e_k^{\mathcal{M}}[\boldsymbol{x}]$ be identical for all $\boldsymbol{x}$ except $\boldsymbol{x}_0$, where $e_k^{\mathcal{S}}[\boldsymbol{x}_0] = 0$ and $e_k^{\mathcal{M}}[\boldsymbol{x}_0] = 1$.

   Consider the following binary hypothesis test to detect motion at location $\boldsymbol{x}_0$:

$$\frac{P(\Psi_k = \psi_k | e_k^{\mathcal{M}})}{P(\Psi_k = \psi_k | e_k^{\mathcal{S}})} \underset{\mathcal{S}}{\overset{\mathcal{M}}{\gtrless}} \theta \frac{P(E = e_k^{\mathcal{S}})}{P(E = e_k^{\mathcal{M}})},$$

   where θ is a constant.

   (a) Show that, if random variables $\Psi_k[\boldsymbol{x}]$ can be assumed conditionally independent for different $\boldsymbol{x}$ given their labels $e_k[\boldsymbol{x}]$, then the above test simplifies to:

$$\frac{P(\Psi_k[\boldsymbol{x}_0] = \psi_k[\boldsymbol{x}_0] | \mathcal{M})}{P(\Psi_k[\boldsymbol{x}_0] = \psi_k[\boldsymbol{x}_0] | \mathcal{S})} \underset{\mathcal{S}}{\overset{\mathcal{M}}{\gtrless}} \theta \frac{P(E = e_k^{\mathcal{S}})}{P(E = e_k^{\mathcal{M}})}.$$

   (b) Justify modeling observation $\psi_k[\boldsymbol{x}_0]$ by a Gaussian random variable with variance $\sigma_{\mathcal{M}}^2$ in moving areas and $\sigma_{\mathcal{S}}^2$ in stationary areas with $\sigma_{\mathcal{M}}^2 >> \sigma_{\mathcal{S}}^2$.

   (c) Justify modeling labels e_k as a Markov random field with either first- or second-order neighborhood, two-element cliques $\{\boldsymbol{x}, \boldsymbol{y}\}$, and Ising potential:

$$V_{\{\boldsymbol{x},\boldsymbol{y}\}}(e_k) = \begin{cases} 0 & \text{if } e_k[\boldsymbol{x}] = e_k[\boldsymbol{y}] \\ 1 & \text{if } e_k[\boldsymbol{x}] \neq e_k[\boldsymbol{y}] \end{cases}$$

   (d) Show that with the above models this binary hypothesis test simplifies to:

$$\psi_k^2[\boldsymbol{x}_0] \underset{\mathcal{S}}{\overset{\mathcal{M}}{\gtrless}} 2\sigma_{\mathcal{S}}^2 \left(\ln(\theta \frac{\sigma_{\mathcal{M}}}{\sigma_{\mathcal{S}}}) + \frac{Q_{\mathcal{S}} - Q_{\mathcal{M}}}{T} \right),$$

   where $Q_{\mathcal{S}}$ is the number of stationary labels in the neighborhood of $\boldsymbol{x}_0$ and $Q_{\mathcal{M}}$ is the number of moving labels in the same neighborhood, and T is the natural temperature of the Gibbs distribution describing MRF E_k.

   (e) In a *Matlab* implementation, $Q_{\mathcal{S}}$ and $Q_{\mathcal{M}}$ must be evaluated at every pixel several times (iterative algorithm due to Markov assumption). Therefore, the efficiency in calculating both values is paramount. Clearly, the use of an *if* statement would be very inefficient. Derive expressions for $Q_{\mathcal{S}}$ and $Q_{\mathcal{M}}$ in terms of realizations e_k for the first-order as well as second-order neighborhood system that could be efficiently implemented in *Matlab* without *if* statements.

2. *Motion detection – Matlab* (50 points).

   (a) Implement in *Matlab*, using matrix/vector operations as much as possible, three variants of the motion detection algorithms from problem 1:

      i. fixed-threshold hypothesis test ($T \to \infty$),
      ii. variable-threshold hypothesis test with first-order MRF,
      iii. variable-threshold hypothesis test with second-order MRF.

      Assume θ=1 and $\sigma_{\mathcal{M}}/\sigma_{\mathcal{S}}$=5. First, select values of $\sigma_{\mathcal{S}}^2$, for each of the image pairs below, that give satisfactory fixed-threshold detection results. Then, with the selected value for $\sigma_{\mathcal{S}}^2$ select value of T that gives satisfactory results for variable-threshold first-order neighborhood detection. Finally, apply the second-order neighborhood algorithm with the same set of parameters.

   (b) Apply each test to the following image pairs that can be found on the course web site under "Assignments/Images":

      i. *Miss America*: frames 1 and 50,
      ii. *Coastguard*: frames 90 and 95,
      iii. *Container*: frames 1 and 30.

   (c) Once you are satisfied with the detection results above, apply each of the three motion detection algorithms above to a 10–15 second sequence of images captured by either network camera: `http://vsn1-iss.bu.edu` or `http://vsn2-iss.bu.edu`. Unfortunately *Matlab* does not support a direct video capture from a URL, so you need to find a method to capture camera's video stream into a file and then read it into *Matlab*. Two options that worked for me under Windows 10 are:

      - A simpler solution is to use FFMPEG (`ffmpeg.org`), but there is no GUI and you need to run commands in shell/command window. However, it works like a charm. To start recording, just type in command window:
        `ffmpeg -i rtsp://vsn1-iss.bu.edu/mpeg4/media.amp?  -acodec copy -vcodec copy C:/filename.mp4`
        and stop recording by pressing Control-C. Errors about incorrect syntax and timestamps will be printed on the screen, but it works well. You can read this file directly into *Matlab* using VideoReader.
      - A partial solution, depending on your operating system and version of VLC (`www.videolan.org`), is to first capture MJPEG stream from the following URL: `http://vsn1-iss.bu.edu/axis-cgi/mjpg/video.cgi?` using VLC (open the stream, play it and capture it with the record button) and then to convert the saved file to MPEG-4 using Convert/Save command. You need to convert to MPEG-4 ("Video - H.264 + MP3 (MP4)") since *Matlab*'s VideoReader cannot read MJPEG. This worked on VLC 2.2.0 under Windows 10, and also under Linux on VLC 2.2.4. It did not work on VLC 2.2.6 under Windows 10.

      If you find other solutions please post them to the Discussion Forum.

   (d) Print the detection results for each algorithm and each image pair above (use `subplot` to include the original frame and three results on a single page for each image). Include your *Matlab* code. Upload the video sequence you captured and three black-and-white label sequences you computed as a **single ZIP file** to the Assignments section on the course web site. Comment on the results obtained.