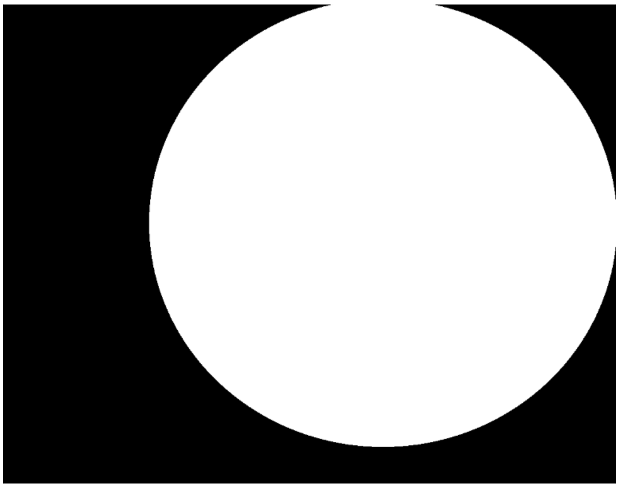
Neurophotonics HW3

Chemda Wiener and Talia Segal

1. Ask User For Recording Path

2. Determine an ROI (plot one frame then ask the user to draw a circle in it)



3. Calculate Read Noise Matrix (read noise in each window)

החישוב בקוד

4. Calculate Pixels-Non-Uniformity ($\sigma sp$) ◦ Average at least 500 frames (rec with laser), subtract background ◦ Calc variance in each window (std_filter^2)

החישוב בקוד

5. Calculate G[DU/e]

חישבנו לפי מה שלמדנו בתרגול:

$$G_{base}\left[\frac{\mathrm{DU}}{\mathrm{e}}\right] = \frac{2^N[DU]}{\text{saturation capacity}\,[e]}\ ; \qquad \mathrm{G_{in}} = 10^{\mathrm{g_{in}}[dB]/20}$$

$$G\left[\frac{\mathrm{DU}}{\mathrm{e}}\right] = G_{base}\cdot G_{in}$$

וקיבלנו ש:

G [DU/e]: 6.1826

6. For every frame subtract the background

החישוב בקוד

7. Calc Raw contrast per window

החישוב בקוד

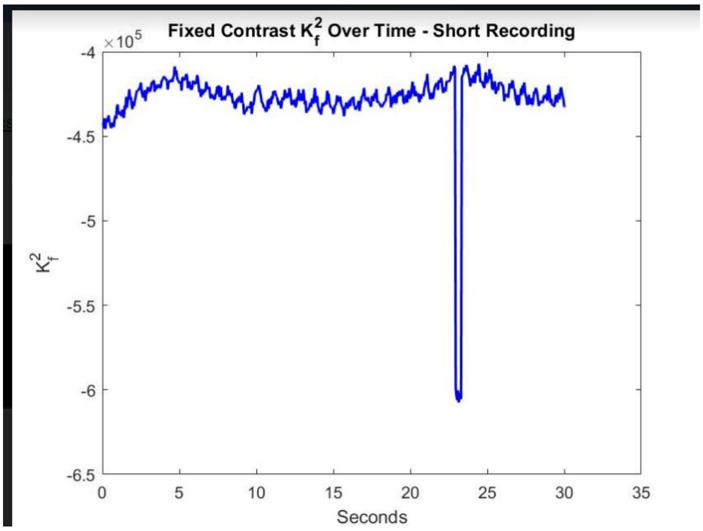
## 8. Calc Fixed contrast per window

$$K_f^2 = K_{raw}^2 - K_s^2 - K_r^2 - K_{sp}^2 - K_q^2$$

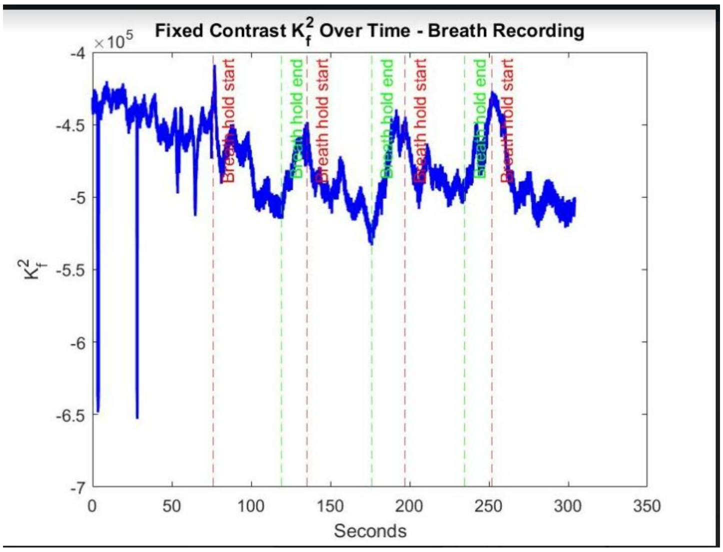
$$K_s^2 = \frac{g}{\langle I \rangle}, K_r^2 = \frac{\sigma_r^2}{\langle I \rangle^2}, K_{sp}^2 = \frac{\sigma_{sp}^2}{\langle I \rangle^2}, K_q^2 = \frac{1}{12\langle I \rangle^2},$$

החישוב בקוד

9. Plot $Kf$ 2 over time for the two recordings




9+10. For the recording with the breath experiment add markup (to the plot) where the breath hold started and ended.



Whole code:

```
% submitted the mask that was drawn with the help of fiji photo
mask_path = "C:\Users\tlsgl\Documents\ קורסים הנדסה\photonic_hw\ex3\Mask.tiff";
mask_tiff = Tiff(mask_path, 'r');
mask = read(mask_tiff);

%% q3
% find the read noise
read_noise_path = 'D:\talia_chemda\read_noise_Gain24_expT0.021ms';
read_noise = Tiff2Matrix(read_noise_path);
read_noise_matrix = stdfilt(read_noise, true(7, 7)).^2;
mean_read_out_noise = mean(read_noise_matrix,3);

%% q4
% calculate the pixels non uniformity - we used the laser regular recording
% and applied it for both breath and regular recordings in future analysis
background_path = 'D:\talia_chemda\background_gain24_expt15ms_chemda';
path = 'D:\talia_chemda\fr20hz_gain24_expt15ms_chemda';
background_rec = Tiff2Matrix_2(background_path);
rec = Tiff2Matrix_2(path);
meanFrameBackground = mean(background_rec, 3);
meanFrame = mean(rec, 3) - meanFrameBackground;
windowSize = 7;
pixels_non_uniformity = stdfilt(meanFrame, true(windowSize)).^2;
```

```
%% q5

% Inputs

N = 12;

saturation_capacity = 10500;

g_in_dB = 24;

% calculate G

G_base = 2^N / saturation_capacity;

G_in = 10^(g_in_dB / 20);

G = G_base * G_in;

disp(['G [DU/e]: ', num2str(G)]);



%% q6 + q7 + q8

Kq = 1 / (12);

% the short recording analysis

files_path_laser = 'D:\talia_chemda\fr20hz_gain24_expt15ms_chemda';

files_laser = dir(strcat(files_path_laser, '\*.tiff'));

for i = 1:length(files_laser)

  t = Tiff(strcat(files_path_laser, '\', files_laser(i).name), 'r');

  imageDataLaser = read(t);

  % subtract background from each frame

  imageDataLaser = imageDataLaser - uint16(meanFrameBackground);

  mean_window = double(imboxfilt(imageDataLaser, 7));

  mean_window = mean(nonzeros(mean_window));

  % calculate the raw contrast per window

  raw = (stdfilt(imageDataLaser, true(7, 7)));

  % calculate fixed contrast per window
```

```
  Kf = (raw - (mean_window .* G) - mean_read_out_noise - pixels_non_uniformity
- Kq) ./ (mean_window.^2);

  % extract only the data within the ROI - mask area

  Kf = Kf .* double(repmat(mask,[1 1 size(imageDataLaser, 3)]));

  Kf_matrix_laser(i) = mean(nonzeros(Kf(:)));

end


% the breath recording analysis

files_path_breath =
'D:\talia_chemda\fr20hz_gain24_expt15ms_chemda_breath';

files_breath = dir(strcat(files_path_breath, '\*.tiff'));

for i = 1:length(files_breath)

  t = Tiff(strcat(files_path_breath, '\', files(i).name), 'r');

  imageData = read(t);

  % subtract background from each frame

  imageData = imageData - uint16(meanFrameBackground);

  mean_window = double(imboxfilt(imageData, 7));

  mean_window = mean(nonzeros(mean_window));

  % calculate the raw contrast per window

  raw = (stdfilt(imageData, true(7, 7)));

  % calculate fixed contrast per window

  Kf = (raw - (mean_window .* G) - mean_read_out_noise - pixels_non_uniformity
- Kq) ./ (mean_window.^2);

  % extract only the data within the ROI - mask area

  Kf = Kf .* double(repmat(mask,[1 1 size(imageData, 3)]));

  Kf_matrix_breath(i) = mean(nonzeros(Kf(:)));

end


% plot for the short recording
```

```
time_per_frame = 1/20;

% convert to seconds for x axis

time_seconds = [1:length(Kf_matrix_laser)] * time_per_frame;

plot(time_seconds, Kf_matrix_laser, 'b-', 'LineWidth', 1.5);

xlabel('Seconds');

ylabel('K_f^2');

title('Fixed Contrast K_f^2 Over Time - Short Recording');


% plot for breath recording

time_per_frame = 1/20;

% convert to seconds for x axis

time_seconds = [1:length(Kf_matrix_breath)] * time_per_frame;

plot(time_seconds, Kf_matrix_breath, 'b-', 'LineWidth', 1.5);

% label when it appears in the plot the breath hold started and ended

% (specifically did not use the times we calculated because I wanted to

% show it from the plot)

xline([76.3,135.35, 197.25, 252.1],'--r','Breath hold start');

xline([119.1,176.25, 234.6],'--g','Breath hold end');

xlabel('Seconds');

ylabel('K_f^2');

title('Fixed Contrast K_f^2 Over Time - Breath Recording');
```