Document for SIQCM\_Theory

Main Function list

**BgNormalization:**

**Frames(i) = (frames – mean(i))\*(meanStd./std(i)) +mMean;**

%Aim Bg normalization for all the frames to Normalize same mean

%and std value

% Input: frames (2d\*1d phase/onri matrix) Original Frames

% Output: Frames (3d matrix) Normalized Frames

**OTF = OTFgenerate(size, kcutoff, k0)**

**% generate OTF function with given cutoff frequency and k vector shift**

%OTF = exp(-0.44\*k/kcutoff);

%size of the OTF supposed to generate.

%Kcutoff cutoff frequency of the OTF, where

%k0 shift vector of the center component

**freq = SeparatedComponents2D(phase, frames),**

**freq = SeparatedComponents2DSIM(phase, frames)**

**% Separate frequency component with given phase values and frames of same orientation (same modulation k vector);**

%For SIQCM

% M = [0.375 0.25\*MF\*exp(-1i\*phase(1)) 0.25\*MF\*exp(1i\*phase(1)) 0.0625\*MF^2\*exp(-2\*1i\*phase(1)) 0.0625\*MF^2\*exp(2\*1i\*phase(1)); … ];

%For SIM

%M = 0.5\*[1 0.5\*MF\*exp(-1i\*phase(1)) 0.5\*MF\*exp(1i\*phase(1)) ;

… … … ];

%Input: phase: array of phase value(R)

% frames: matrix of frames of same orientation

%Output: freq : frequency component, [0 +1 -1 +2 -2] for SIQCM and [0 +1 -1] for SIM

**[Fsum, Fsum2, Fperi, Fcent] = WienerF(f, k, kcutoff, gamma)**

**% Apply Wiener filter and combine the frequency component in frequency space**

% Input: f the matrix of frequency component

% k, modulation wave factor

% kcutoff cutoff frequency

%Output: Fsum the sum of all frequency component

% Fsum2 the same above by without triangular mask

% Fperi sum of non-center frequency component

**Fsum = WienerF2(f, k, kcutoff, gamma)**

**%Apply Wiener filter first and combine the frequency component in real space**

% Input: f the matrix of frequency component

% k, modulation wave factor

% kcutoff cutoff frequency

%Output: Fsum the sum of all frequency component

**w = WienerMask(t, kcutoff, k, korder, gamma)**

**% Generate the shifted Wiener filter mask for each individual component**

%Input: t the mask size size\*size

% k, modulation wave factor

% kcutoff cutoff frequency

% korder the vector Wiener filter supposed to shift

**H = triMask(size, kmax, k0, kshift)**

**% Generate the shifted triangular mask at given size**

%Input: size the mask size size\*size

% kmax cutoff frequency

% k0 the mean value of modulation wave factor

% kshift the vector Wiener filter supposed to shift