

- Step 1: Calculate shift x,y pairs for each frame
- Step 2: apply shift to movie frames

```
% Reads a TEM movie file or image sequence, computes shift of image vs first frame, corrects b
y wrapping or padding, saves as uncompressed avi
% Josh Sugar and Dave Robinson, Sandia National Labs, (c) 2010-2011

InputForm = questdlg('What is your input file format?','Input format','PNG','TIF','Movie','Mov
ie');

inmovie=0;
if strcmp(InputForm,'PNG',3);
    filetype = 'png';
elseif strcmp(InputForm,'TIF',3);
    filetype = 'tif';
else
    filetype = 'Movie';
    inmovie=1;
end

% Image sequence can be generated from quicktime pro.
% TIFF, when uncompressed, runs faster in both quicktime and matlab.
% TIFF Packbits compression in quicktime doesn't do much for TEM images.
% PNG uses lossless compression, so is more efficient with disk space,
% especially useful if you run quicktime pro on another computer and
% transfer files via USB drive.
% Quicktime tends to choke when reassembling long image sequences into a movie.
% This version generates uncompressed AVI output.

% For movies:
% Containers supported: .avi, .mpg, .mj2 (extremely obscure)
% Windows only: .wmv, .asf, .asx
% Mac only: .mp4, .mov
% Codecs supported: good luck - generally only obsolete ones.
% Microsoft DirectShow codecs include:
% Cinepak, DV, H.264 (MS MP2), ISO MP4 1.0, MS MP4 v3, MJPEG, MP1, MP2

% Bug: sometimes the shifts are off by one width or height, resulting in
% excessive padding in 'pad' mode (innocuous in 'wrap' mode).

% x domain shift f(x-a) transforms to F(w)exp(-iaw)
% Find a, given f(x) and f(x-a)
% F(w)*conj(F(w)exp(-iaw)) gives |F(w)|^2*exp(iaw) (keeping shift but discarding phase info in
image)
% ifft of |F(w)| is maximum at zero because all components are in phase there.
% ifft of the whole thing is shifted by a, so identify the location of the maximum.
```

Step 1: Calculate shift x,y pairs for each frame

```
% Open File
if inmovie;
```

```

[moviefilename,inputpath]=uigetfile('*.','Choose a movie file to drift correct (cancel skips to next step)');

else % image sequence
    inputpath=uigetdir('*.','Choose a folder containing the image sequence (cancel skips to next step)');
    moviefilename = '';
end

if inputpath~=0; % skip to step 2 if canceled

    tic; % start stopwatch

    moviefile=fullfile(inputpath,moviefilename);

    % Set up input file
    if inmovie;
        readobject=mmreader(moviefile);
        get(readobject);
        % Get number of frames
        if isempty(readobject.NumberofFrames)==1;
            disp('I need to read through this variable bitrate movie to count the frames.')
            read(readobject,inf);
        end
        nFrames = readobject.NumberOfFrames;

    else % image sequence
        files=dir(fullfile(inputpath, strcat('*.','filetype')));
        nFrames = length(files);
    end % inmovie

    shifty = zeros(nFrames,1);
    shiftx = zeros(nFrames,1);

    % Get reference frame (first one in the sequence)

    if inmovie;
        frameref=read(readobject,1);
    else % image sequence
        frameref=imread(fullfile(inputpath,files(1).name),filetype);
    end %inmovie

    imageref=frameref(:,:,1);
    fft_ref=fft2(imageref);

    [vidHeight vidWidth blank] = size(frameref); % The blank variable here gets rid of extra padding !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
    centery=(vidHeight/2)+1;
    centerx=(vidWidth/2)+1;

    wb=waitbar(0,'Please Wait... Calculating Drift...');
    for i=1:nFrames;
        if inmovie;
            imagei=read(readobject,i);
        else % image sequence
            imagei=imread(fullfile(inputpath,files(i).name),filetype);
        end % inmovie
    end
end

```

```

    framei=framei(:,:,1);
    fft_frame=fft2(framei);
    prod=fft_ref.*conj(fft_frame);
    cc=ifft2(prod);
    [maxy,maxx]=find(fftshift(cc)==max(max(cc)));
    % fftshift moves corners to center; max(max()) gives largest element; find returns indices
    of that point

    shifty(i)=maxy(1)-centery;
    shiftx(i)=maxx(1)-centerx;
    % previous version didn't subtract center point here

    if mod(i,500)==0;
        disp(sprintf('%d frames of %d done',i,nFrames))
        waitbar(i/nFrames,wb);
    end
end % i loop
close(wb);
% save results to file
shiftdatafilename=fullfile(inputpath, strcat(moviefilename, '_driftdata.mat'));
if inmovie;
    inputpath = moviefile;
end
save(shiftdatafilename, 'nFrames', 'shifty', 'shiftx', 'inputpath', 'filetype');

disp('I finished calculating the drift data. It was saved to:')
disp(shiftdatafilename)
toc % report time elapsed
load gong
player=audioplayer(y,Fs);
play(player);
end % if didn't cancel drift correct

```

Step 2: apply shift to movie frames

```

clear all
% Get parameters and input file from user

shiftmode = questdlg('Pad frame with empty space, or wrap within same bounds?', 'Pad or Wrap', '
Pad', 'Wrap', 'Wrap');

avgframes = str2num(questdlg('Average together how many frames?', 'Averaging', '5', '3', '1', '1'))
;

bigIter = str2num(questdlg('Max number of input frames per output file', 'Output file size', '15
000', '5000', '2000', '2000'));

[shiftfilename, shiftpathname]=uigetfile('*.mat', 'Please locate the drift data for the movie to
correct...');

if shiftfilename~=0; % skip to end if canceled

tic % start stopwatch

shiftfile=fullfile(shiftpathname, shiftfilename);
load(shiftfile);

```

```

inmovie = 0;
if strcmp(filetype,'Mov',3);
    inmovie = 1;
end

% Read folder to get basic info

if inmovie;
    readobject=mmreader(inputpath);
    get(readobject);
    vidHeight = readobject.Height;
    vidWidth = readobject.Width;
    framerate = readobject.Framerate;

else % image sequence
    files=dir(fullfile(inputpath, strcat('*. ', filetype)));
    nFrames = length(files);
    framerate = 29.97;

    % get height and width from first image
    frameref=imread(fullfile(inputpath,files(1).name),filetype);
    frameref=uint8(frameref(:,:,1));
    [vidHeight vidWidth] = size(frameref); % The blank variable here gets rid of extra padding !
    !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
end %inmovie

if strcmp(shiftmode,'Pad',3);
    newsizey=2*max(abs(shifty))+vidHeight;
    newsizex=2*max(abs(shiftx))+vidWidth;
    % assumes max positive shift = max negative shift; centers reference frame

    midindexy=(newsizey-vidHeight)/2+1;
    midindexx=(newsizex-vidWidth)/2+1;
else
    % wrap
    newsizey=vidHeight;
    newsizex=vidWidth;
    midindexy=1; % these won't be used
    midindexx=1;
end

for j=1:floor(nFrames/bigIter)+1;
    disp(sprintf('\nMovie file %d\n',j))
if floor(nFrames/bigIter)+1==1;
    k=1;
    bigIter=nFrames;
end
if j==1;
    k=1;
elseif j==floor(nFrames/bigIter)+1;
    k=(j-1)*bigIter+1;
    bigIter=mod(nFrames,2000);
else
    k=(j-1)*bigIter+1;
end

% Create array of 8-bit grayscale frame structures

```

```

clear mov_shift
numofframes = floor(bigIter/avgframes)-1;
mov_shift(1:numofframes)=struct('cdata',zeros(newsizey,newsizey,1,'uint8'),...
    'colormap',[linspace(0,1,256)',linspace(0,1,256)',linspace(0,1,256)']);
disp('I allocated the new movie structure...')

% Process each frame
for i = 1:numofframes;
    frame_sum=zeros(newsizey,newsizey,'uint16');
    for z = 1:avgframes;
        zz=k+(i-1)*avgframes+z-1;
        if zz>nFrames;
            continue;
        end
        if inmovie;
            imagei=read(readobject,zz);
        else
            imagei=imread(fullfile(inputpath,files(zz).name),filetype);
        end
        framei=uint8(imagei(:,:,1));
        if strcmp(shiftmode,'Pad',3);
            frame_shift=zeros(newsizey,newsizey,'uint8');
            frame_shift(midindexy+shifty(zz):midindexy+shifty(zz)+(vidHeight-1),...
                midindexx+shifty(zz):midindexx+shifty(zz)+(vidWidth-1))=framei;
            frame_sum=frame_sum+uint16(frame_shift);
        else
            % Wrap
            frame_sum=frame_sum+uint16(circshift(framei,[shifty(zz) shifty(zz)]));
        end % if pad or wrap
    end % z loop
    mov_shift(i).cdata=uint8(round(frame_sum/z));
    if mod(i,500)==0;
        disp(sprintf('%d frames of %d done',i,numofframes))
    end
end % i loop
k=k+bigIter;

% Write Movie to new AVI File
disp('I am writing the movie now...')
if inmovie;
    [moviepath,moviename,movieext]=fileparts(inputpath);
    filewritename=strcat(fullfile(moviepath,moviename),'_',num2str(j),'.avi');
else % image sequence
    filewritename=strcat(fullfile(inputpath,'corrected'),'_',num2str(j),'.avi');
end % inmovie
movie2avi(mov_shift,filewritename,'fps',framerate,'compression','None')

end % j loop (bigIter blocks)

disp('The corrected movies are complete ...enjoy...')
toc % report time elapsed
load gong
player=audioplayer(y,Fs);
play(player);
end % if didn't cancel dialog box

```

