ACTIVE OPTICAL SYSTEMS

**AN040**

# Fitts Correlation Study

**Author:** **Dr. Justin Mansell and Brian Henderson**
**Revision:** **10/23/12**

## 1 Introduction

One of the earliest descriptions of the Fitts Correlation technique is described in U.S. Patent 4,133,004 entitled "Video Correlation Tracker". The appeal of the technique is that it produces good accuracy results with less computational intensity than a full correlation calculation. The patent is a great reference, but a recent presentation by Shai Levi and Noam Helmer entitled "Fitts Correlation Tracker simulation" gives an even more concise tutorial. Given a noisy signal y of variable x with an offset δ, we can write the total signal as a sum of the noise, n, and pure signal, s, terms as,

$$y(x) = s(x-\delta) + n(x)\,.$$

The Fitts correlation tracking algorithm shows that you can estimate the signal offset δ by

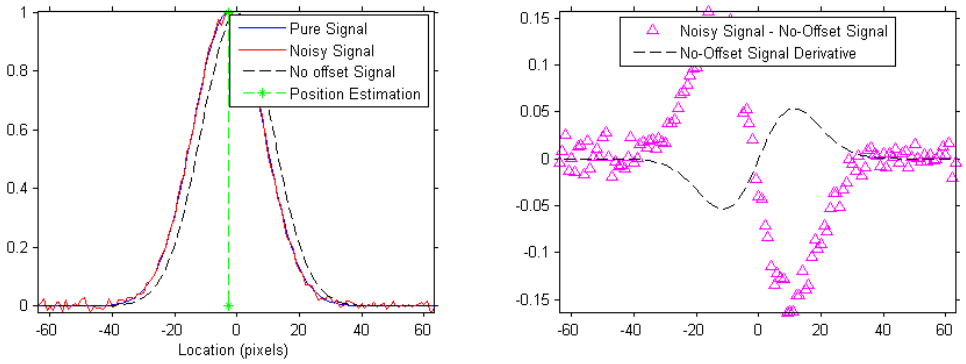
$$\hat{\delta} = \frac{1}{C}\int_a^b W(x)\cdot\left[y(x) - s(x)\right]$$

*where*

$$W(x) = -\frac{\partial s(x)}{\partial x} \text{ and } C = \int_a^b W^2(x)dx$$

This form is useful as-is if the signal is a known shape. To investigate this, we chose to simulate the tracking of a Gaussian in 1D. The code for this is in the appendix. The Gaussian we used had a 16 pixel sigma (radius at 1/e amplitude). The frames were 128 pixels across. The entire frame was used for the integration. The noise was the same for each frame and was Gauss-normalized to 1% amplitude relative to the peak. Figure 1.1 illustrates the signals and the intermediate calculation results when doing the Fitts estimation for a small offset. When examining the form of the difference signal and the derivative, we see that they are indeed the same shape. The offset is estimated by doing an overlap-integral between the difference and derivative, so it is, in a sense, determining the amount of the difference signal that is in the derivative (or vice-versa). Since the intensity pattern of any image is essentially made up of a weighted sum of the point spread functions of the imaging system and the point-spread functions are often very similar in shape to a Gaussian (sinc or Airy patterns), this example illustrates how the Fitts correlation technique is often quite accurate at estimating the signal offset.


© Active Optical Systems, LLC




(a) (b)

**Figure 1.1: (a) Gaussian Signals and Fitts Location Estimate; (b) Pure Signal Derivative and Difference between the Noisy Signal and the No-Offset Signal**

Figure 1.2 shows the results when scanning over varying offsets. For our Gaussian signal, the position estimation or offset estimation is quite linear around the center of the Gaussian, but has more error as the offset is larger. For our case, this technique appears to break down when the input offset is more than half the Gaussian sigma (±8 pixels). This indicates that we need to be fairly close to the center to achieve good accuracy of the Fitts algorithm, but over this offset range, the sign of the error function remains good throughout.

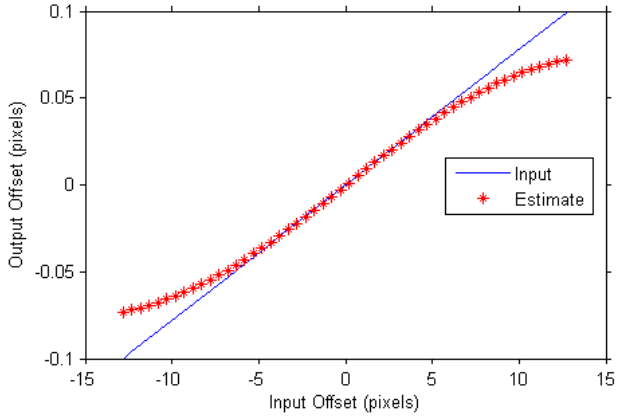



**Figure 1.2 - Results of the Fitts calculation with a 1D noisy Gaussian signal**

Figure 1.3 shows the results when varying the input signal offset over the entire window. Over this range, we see that the ultimate breakdown of the estimate occurs when the offset equals the Gaussian 1/e (or sigma) radius. Beyond this point, the estimator begins reporting values that are not tracking the offset.


© Active Optical Systems, LLC

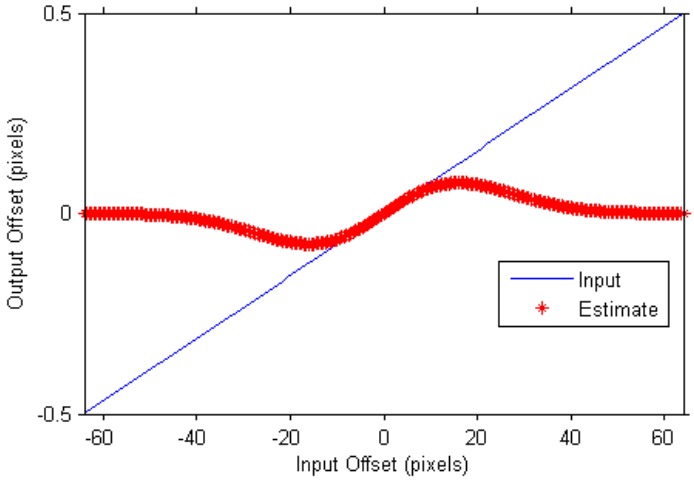



**Figure 1.3 - Fitts results over a larger offset range**

## 1.1 2D Fitts Estimation with a Known Signal

We extended the calculation to a 2D image with the code in the appendix. The offset was applied equally in each direction. The results were essentially the same except the break in the error function was closer to 16/√2 = 11.3.

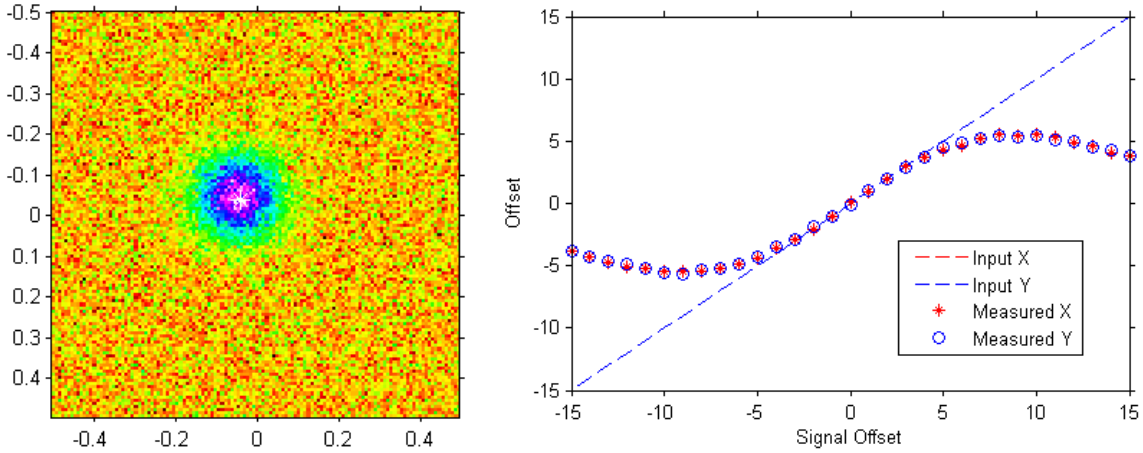



**Figure 1.4 (a) An Example Image; (b) Results**

We next evaluated the spatial frequency response by creating a 1D sinusoidal signal across the 2D image. We varied the spatial frequency over the 128x128 image from 1 to 10 cycles over the aperture. The code is in the appendix. Figure 1.5 shows the Fitts position estimation for varying spatial frequencies. The results are exactly what could be expected from the Gaussian study results. The higher spatial frequency results show breakdown in the signal earlier from the zero offset position than the lower spatial frequency results. The 10 cycles per aperture case had a 12.8 pixel spatial period. The break point of this signal was ±3.25 pixels corresponding to a phase angle of ±90 degrees.


© Active Optical Systems, LLC

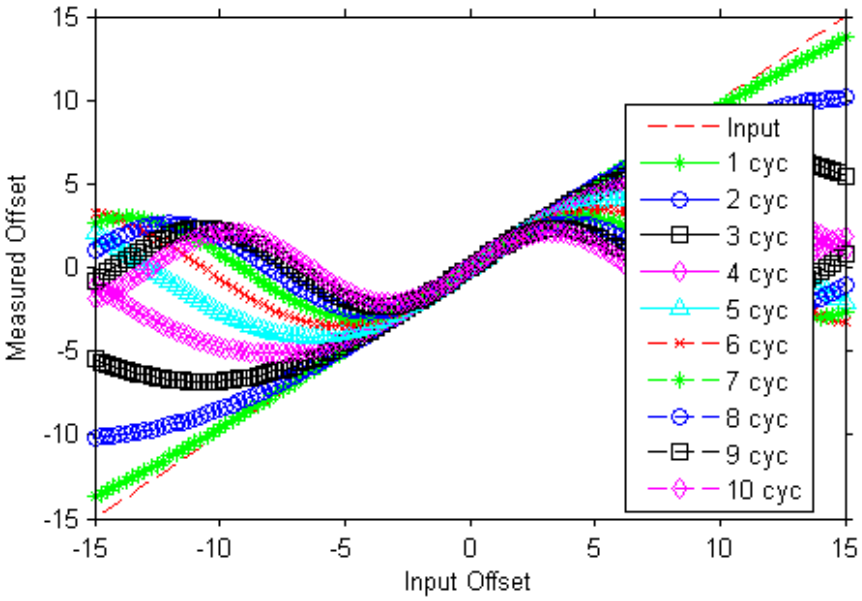



**Figure 1.5 - Fitts Position Estimation for Varying Spatial Frequencies**

## 1.2 2D Fitts Estimation with a Windowed Noisy Gaussian Signal

We next analyzed the case where we move a small 32x32 window over the 128x128 image using feedback from the Fitts calculation over that window. We also compared the results of the Fitts calculation to a thresholded centroid calculation. We used a 20% threshold for these calculations. Again, the Matlab code for this is in the appendix. Figure 1.6 shows some of the image results for one of the Fitts estimations. The dotted white line in the bottom right image shows the window on the full 128x128 image.

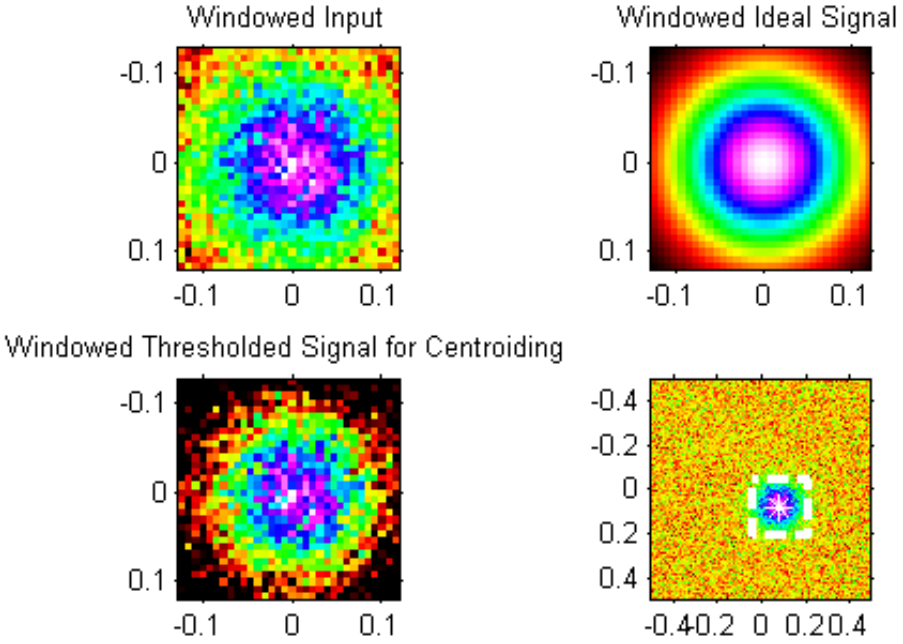
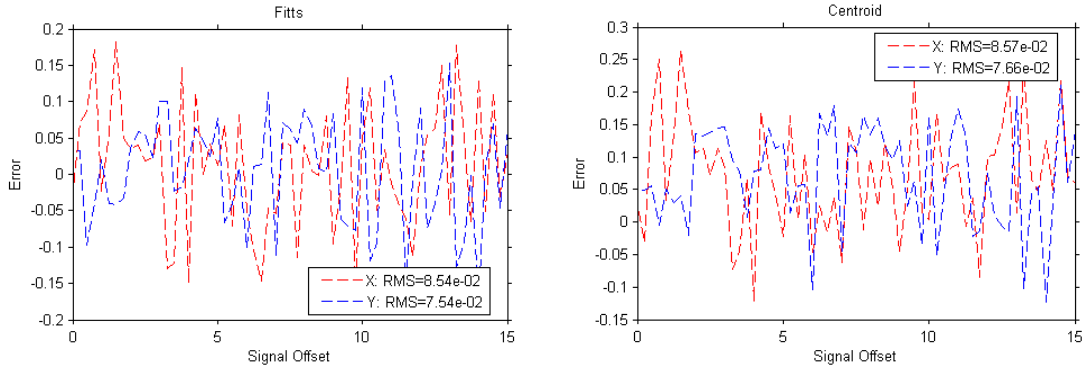



**Figure 1.6 - (bottom right) Full Noisy Gaussian Signal; (top left) Windowed Portion of the Image for the Fitts Estimation; (top right) Windowed Ideal Signal for the Fitts Estimation; (bottom left) Thresholded Intensity for the Centroid Calculation**


© Active Optical Systems, LLC

Figure 1.7 shows the error in the results of the Fitts estimation and the thresholded centroid operations. The legend indicates the RMS of the error for both of these is very similar (around 8% of a pixel). These calculations were done with pixelization, but not any digitization on the intensity.




(a) (b)

**Figure 1.7 - Error of the Fitts (a) and Centroid (b) calculations over a range of pixel motions**

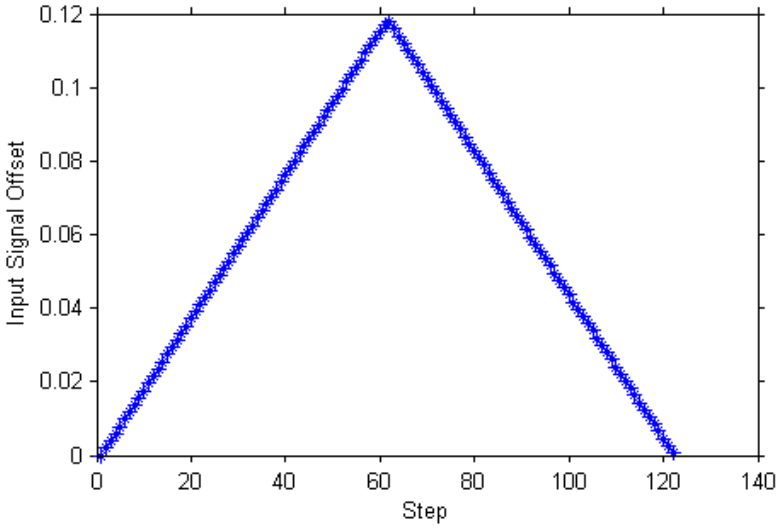
## 1.3 2D Fitts Estimation with a Unknown Signal (Map Updates)

In the case of an unknown signal, the signal needs to be derived from the camera image. This can be accomplished with a weighted average of the previous mask and the new measured signal that is typically of the form,

$$mask_{n+1} = (1-w)\cdot y + w \cdot mask_n ,$$

where w is a weighting factor that is multiplied by the new mask to incorporate and y is the new measured signal. With this system, the We chose to begin with the first frame as the initial mask, but other algorithms, like a direct average of the first set of frames would likely produce better results. We found it was important to move the mask based on the Fitts estimation before calculating the new mask.

We again generated a 128x128 image and did the Fitts estimation over a 32x32 window. In this case we scanned the spot from the center out 15 pixels in quarter-pixel increments. We also had a small offset to the backwards scan of the beam to distinguish it from the forward scan of the step size divided by three. Figure 1.8 shows the input signal offsets.





© Active Optical Systems, LLC

**Figure 1.8 – Simulation Input Signal Offsets**

Figure 1.9 illustrates a frame near the maximum displacement of the Gaussian spot. The Gaussian spot radius was smaller than prior studies at 1/80$^{th}$ of the full 128-pixel frame diameter. The weighting factor was 0.005. The Gauss-normalized noise magnitude was extremely small at 0.0003. The Gaussian signal maximum amplitude was 1.0. We also performed a thresholded centroid for comparison with the threshold set to 20%. The Matlab code for this simulation is in the appendix.

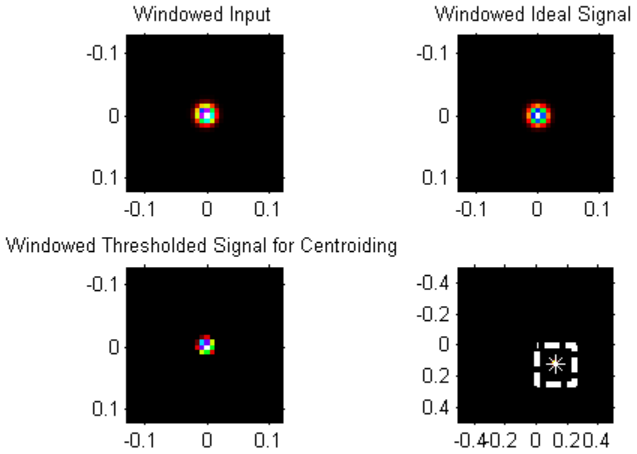
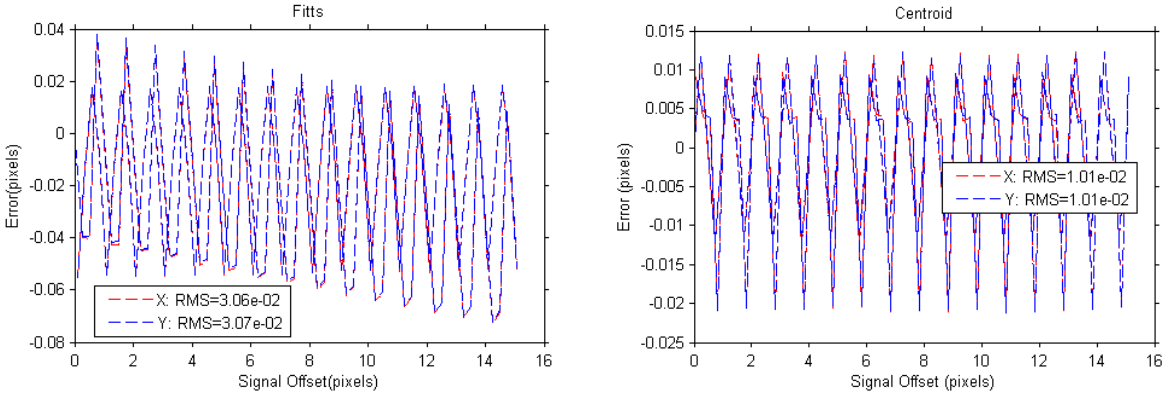



**Figure 1.9 - (top left) is the windowed input signal; (top right) is the mask (aka the ideal signal); (bottom left) is the thresholded input used for centroid calculation; (bottom left) is the full measured signal with Fitts location (white asterisk) and clip window (white dotted line).**

Figure 1.10 shows the error in the estimation for thresholded centroid and a Fitts estimation. The Fitts estimation indicated a 3% RMS error and the thresholded centroid was only 1% in this case. Fitts is generally better for signals with structure, like tracking real objects, but not as good for clean Gaussian-like signals.




(a) (b)

**Figure 1.10 (a) shows the error from the Fitts estimation; (b) shows the error from the thresholded centroid estimation.**


© Active Optical Systems, LLC

### 1.3.1 Adding More Noise

We ran the simulation again with a noise amplitude of 0.05 and a map average weighting factor of 0.1. Figure 1.11 shows an example of the same images as Figure 1.9 with the increased noise. The added noise clearly makes the signal very weak.

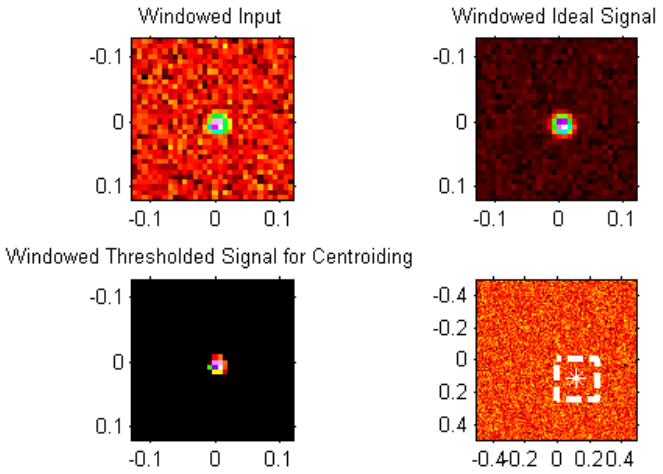
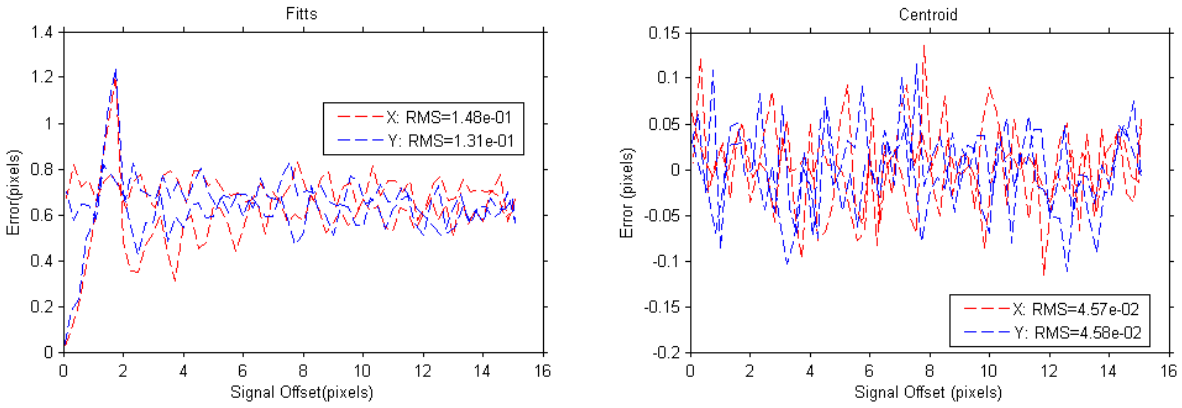



**Figure 1.11 - Example Results with More Noise**

Figure 1.12 shows the estimation error of the Fitts and thresholded centroid routines with the increased noise. Fitts and thresholded centroid had RMS estimation errors of approximately 14% and 5% respectively. There is more error in the initial frames of the Fitts estimation than the final frames due to the evolution of an improved mask. The Fitts routine had an offset in both axes of 0.6 pixels.




(a) (b)

**Figure 1.12 (a) and (b) are the Error of the Fitts and Thresholded Centriod Calculations respectively.**

Decreasing the weighting factor to 0.05 reduced the error slightly (14.4% and 12.3% in X and Y respectively), and showed a slight decrease in the offset (0.45). Further decrease in the weighting factor to 0.005 caused the Fitts algorithm to no longer track the spot.


© Active Optical Systems, LLC

### 1.3.2 Sinusoidal Signal

We also investigated tracking a signal that was a 2D array of spots generated by a product of two sinusoidal signals. One axis used a spatial frequency two times the other axis. Figure 1.13 shows the evaluation of this signal about half-way through the 15-pixel offset done in quarter-pixel steps. The noise amplitude was 5%. The mask weighting was 5%. By this stage in the tracking, the mask signal in the upper right corner was sufficiently averaged to minimize the impact of noise.

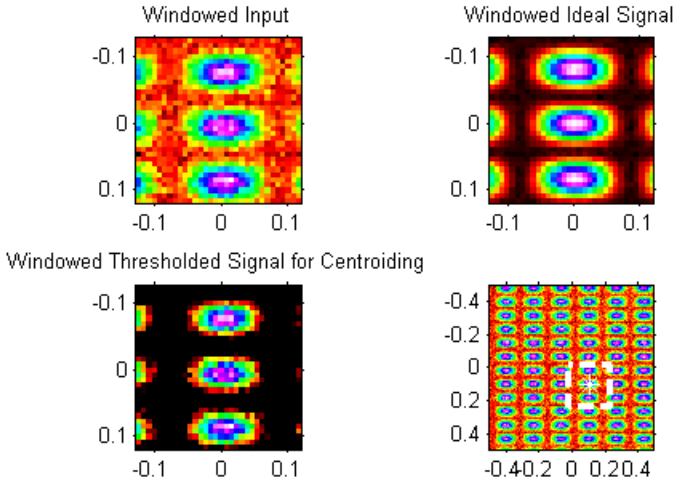
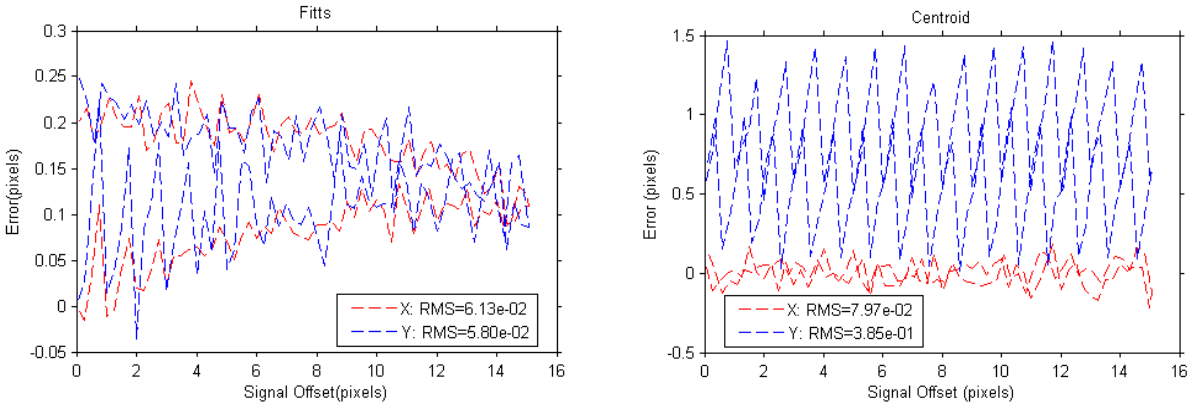



**Figure 1.13 - Non-Gaussian Signal Evaluation**

Figure 1.14 shows the error in the Fitts estimation and the 20% thresholded centroid calculation (evaluated only over the 32x32 pixel window). The Fitts estimation produced around 6% RMS error in both axes, but the centroid estimation had 8% in one axis and 39% in the other axis. This complex signal shows the power of Fitts estimation when the signal has significant structure. The Matlab code for this evaluation is in the appendix.




(a) (b)

**Figure 1.14 (a) and (b) show the error of the signal offset estimation using Fitts and thresholded centroid operations respectively**

## 2 Further Analysis of Fitts Estimation

In the future, we would like to investigate:

- How can we detect if Fitts is failing? Can we look at map changes?


© Active Optical Systems, LLC

- Can we compare Fitts estimation with a true Fourier-transform correlation with and without map updates?


© Active Optical Systems, LLC

# 3 Appendix

Most of the following code will execute within a standard copy of Matlab, but some of the functions are not part of the standard package and will have to be removed. It is our belief that these functions are limited to specialized graphics and PowerPoint interface functions.

## 3.1 1D Fitts Correlation Tracking Code in Matlab

```
% 1D Fitts Correlation testing
close all; clear all; clc;
ppt=1;

Dap = 1.0;
nxy = 128;
w = Dap/8;
dxy = Dap/nxy;
x = (-nxy/2:1:nxy/2-1)*dxy;
noise = 0.01.*randn(1,nxy);
s2 = exp(-1 * ((x)./w).^2);
W = -gradient(s2);
C = sum(W.^2);

nf([  321         363        1020         325]);
c=0;
dxvec = [-Dap/10:dxy/2:Dap/10];
for dx = dxvec;
    c=c+1;
    s = exp(-1 * ((x-dx)./w).^2); %generate pure signal with offset
    y = s + noise; %generate noisy signal
    delta = (1/C .* sum(W.*(y-s2)))*dxy %Fitts estimate
    offsetSave(c)=dx;
    deltaSave(c)=delta;

    clf;
    subplot(1,2,1);
    plot(x./dxy,s);
    hold on;
    plot(x./dxy,y,'r-');
    plot(x./dxy,s2,'k--');
    plot([delta delta]./dxy,[0 max(s2)],'g--*');
    xlabel('Location (pixels)');
    legend('Pure Signal','Noisy Signal','No offset Signal','Position
Estimation','Location','Best');
    axis tight;

    subplot(1,2,2);
    plot(x./dxy,y-s2,'m^')
    hold on;
    plot(x./dxy,W,'k--')
    legend('Noisy Signal - No-Offset Signal','No-Offset Signal
Derivative','Location','Best');
    axis tight;
    drawnow;
    if (ppt)
        ToPPT(gcf,'Summary',[5 5 c],c==1);
    end;
```


© Active Optical Systems, LLC

```
end;

%% summary of results
nf;
plot(dxvec,offsetSave);
hold on;
plot(dxvec,deltaSave,'r*');

nf;
plot(dxvec./dxy,offsetSave);
hold on;
plot(dxvec./dxy,deltaSave,'r*');
xlabel('Input Offset (pixels)');
ylabel('Output Offset (pixels)');
legend('Input','Estimate','Location','Best');

if (ppt)
    ToPPT('Summary of Results');
end;
```

### 3.2 2D Fitts Correlation Script with a Known Gaussian Signal

```
% 2D Fitts Estimation Evaluation/Demonstration
Setup;
plot1D=0;
ppt=1;

Dap = 1.0;
nxy = 128;
w = Dap/10;
dxy = Dap/nxy;
x = (-nxy/2:1:nxy/2-1)*dxy;
[xx,yy] = meshgrid(x,x);
s2 = exp(-1 * ( (xx./w).^2 + (yy./w).^2));
Wy = -gradient(s2);
Wx = -gradient(s2')';
Cx = sum(sum(Wx.^2));
Cy = sum(sum(Wy.^2));

nf;
subplot(1,2,1); show(Wx); title('Wx');
subplot(1,2,2); show(Wy); title('Wy');

nf;
c=0;
dy=1*dxy;
dxv = [-15*dxy:dxy/1:15*dxy];
for dx = dxv;
    noise = 0.1.*randn(nxy,nxy);
    dx
    dy=dx;
    c=c+1;
    s = exp(-1 * ( ((xx-dx)./w).^2 + ((yy-dy)./w).^2) );
    y = s + noise;
    deltax = (1./Cx .* sum(sum(Wx.*(y-s2))))*dxy
```


© Active Optical Systems, LLC

```
    deltay = (1./Cy .* sum(sum(Wy.*(y-s2))))*dxy
    offsetSaveX(c)=dx;
    deltaSaveX(c)=deltax;
    offsetSaveY(c)=dy;
    deltaSaveY(c)=deltay;

    if (plot1D)
        clf; plot(x./dxy,s');
        hold on;
        plot(x./dxy,y','r-');
        plot([delta delta]./dxy,[0 max(s2(:))],'g--*');
    else
        clf;
        imagesc(x,x,y); axis image;
        hold on;
        %plot(deltax,deltay,'w*','MarkerSize',10);
        plot(deltay,deltax,'w*','MarkerSize',10);
    end;
    pause(0.05); drawnow;
    if (ppt)
        ToPPT(gcf,'Data',[5 5 c],c==1);
    end;
end;
%% summary
nf;
plot(dxv./dxy,offsetSaveX./dxy,'r--');
hold on;
plot(dxv./dxy,offsetSaveY./dxy,'b--');
plot(dxv./dxy,deltaSaveX./dxy,'r*');
plot(dxv./dxy,deltaSaveY./dxy,'bo');
xlabel('Signal Offset');
ylabel('Offset');
legend('Input X','Input Y','Measured X','Measured Y','Location','Best');
if (ppt) ToPPT(); end;
```

### 3.3 2D Fitts Correlation with a Sinusoidal Signal

```
%% 2D Fitts Estimation Evaluation/Demonstration - Sinusoidal Signal
Setup;
plot1D=0;
ppt=1;
showImages = 0;

Dap = 1.0;
nxy = 128;
w = Dap/10;
dxy = Dap/nxy;
x = (-nxy/2:1:nxy/2-1)*dxy;
[xx,yy] = meshgrid(x,x);

h1 = nf([50 50 600 600]);
h2 = nf([350 50 600 600]);
periodVec = [1:1:10];
dxv = [-15*dxy:dxy/4:15*dxy];
p=0;
for period = periodVec;
    p=p+1;
```


© Active Optical Systems, LLC

```
%s2 = exp(-1 * ( ((xx)./w).^2 + ((yy)./w).^2) );
s2 = sin(xx * 2.0 * pi / (Dap/period));

figure(h1);
clf;
show(s2); title(sprintf('%i cycles',p));
if (ppt)
    ToPPT(gcf,'Cycles',[3 4 p],p==1);
end;

Wy = -gradient(s2);
Wx = -gradient(s2')';
Cx = sum(sum(Wx.^2));
Cy = sum(sum(Wy.^2));

figure(h1);
subplot(1,2,1); show(Wx); title('Wx');
subplot(1,2,2); show(Wy); title('Wy');

c=0;
dy=1*dxy;
for dx = dxv;
    noise = 0.1.*randn(nxy,nxy);
    dx
    dy=dx;
    c=c+1;
    s = sin((xx-dx) * 2.0 * pi / (Dap/period));
    %s = exp(-1 * ( ((xx-dx)./w).^2 + ((yy-dy)./w).^2) );
    y = s + noise;
    deltax = (1./Cx .* sum(sum(Wx.*(y-s2))))*dxy
    deltay = (1./Cy .* sum(sum(Wy.*(y-s2))))*dxy
    offsetSaveX(p,c)=dx;
    deltaSaveX(p,c)=deltax;
    offsetSaveY(p,c)=dy;
    deltaSaveY(p,c)=deltay;
    if (showImages)
        figure(h2);
        if (plot1D)
            clf; plot(x./dxy,s');
            hold on;
            plot(x./dxy,y','r-');
            plot([delta delta]./dxy,[0 max(s2(:))],'g--*');
        else
            clf;
            imagesc(x,x,y); axis image;
            hold on;
            %plot(deltax,deltay,'w*','MarkerSize',10);
            plot(deltay,deltax,'w*','MarkerSize',10);
        end;
        pause(0.05); drawnow;
        if (ppt)
            ToPPT(gcf,'Data',[5 5 c],c==1);
        end;
    end;
end;
%% summary
```


© Active Optical Systems, LLC

```
    figure(h2);
    plot(dxv./dxy,offsetSaveX(p,:)./dxy,'r--');
    hold on;
    plot(dxv./dxy,offsetSaveY(p,:)./dxy,'b--');
    plot(dxv./dxy,deltaSaveX(p,:)./dxy,GetLineSpec(p*2-1));
    plot(dxv./dxy,deltaSaveY(p,:)./dxy,GetLineSpec(p*2));
    xlabel('Signal Offset');
    ylabel('Offset');
    legend('Input X','Input Y','Measured X','Measured Y','Location','Best');
    drawnow;
end;
if (ppt) ToPPT(); end;

%% final summary
nf;
plot(dxv./dxy,offsetSaveY(1,:)./dxy,'r--');
hold on;
p=0;
leg{1} = 'Input';
for period=periodVec;
    p=p+1;
    plot(dxv./dxy,deltaSaveY(p,:)./dxy,GetLineSpec(p));
    leg{p+1}=sprintf('%i cyc',period);
end;
xlabel('Input Offset');
ylabel('Measured Offset');
legend(leg,'Location','Best');
if (ppt) ToPPT(); end;
```

### 3.4 2D Fitts Correlation – Noisy Gaussian – Small Window

```
% FITTS test based on the report of Shai Levi and Noam Helmer from 2007
Setup;
plot1D=0;
ppt=1;

Dap = 1.0;

%full coordinates
nxy = 128;
w = Dap/10;
dxy = Dap/nxy;
x = (-nxy/2:1:nxy/2-1)*dxy;
[xx,yy] = meshgrid(x,x);

%window coordinates
nxyW = 32;
xW = (-nxyW/2:1:nxyW/2-1)*dxy;
[xxW,yyW] = meshgrid(xW,xW);

nf;
c=0;
dy=0;
dxv = [0:dxy/4:15*dxy];
deltax = 0; deltay = 0;
xc = nxy/2+1 - nxyW/2 + deltax;
```


© Active Optical Systems, LLC

```
yc = nxy/2+1 - nxyW/2 + deltay;
for dx = dxv;

    %% setup the clean signal
    s2 = exp(-1 * ( (xxW./w).^2 + (yyW./w).^2));
    Wy = -gradient(s2);
    Wx = -gradient(s2')';
    Cx = sum(sum(Wx.^2));
    Cy = sum(sum(Wy.^2));
    if (dx==min(dxv))
        nf;
        subplot(1,2,1); show(Wx); title('Wx');
        subplot(1,2,2); show(Wy); title('Wy');
        if (ppt)
            ToPPT('Derivates');
        end;
    end;

    %% setup the noisy measured signal
    noise = 0.1.*randn(nxy,nxy);
    dx
    dy=dx;
    c=c+1;
    s = exp(-1 * ( ((xx-dx)./w).^2 + ((yy-dy)./w).^2) );
    y = s + noise;
    yW = y(xc-1+(1:nxyW),yc-1+(1:nxyW));

    %% do thresholded centroiding
    thr = 0.2;
    yWt = yW - thr;
    yWt = yWt .* (yWt>0);
    deltaxCENTROID = sum(sum(yWt.*yyW))./sum(sum(yWt))
    deltayCENTROID = sum(sum(yWt.*xxW))./sum(sum(yWt));
    deltaSaveXCENTROID(c)=deltaxCENTROID + dxy*(xc-(nxy/2+1)+nxyW/2);
    deltaSaveYCENTROID(c)=deltayCENTROID + dxy*(yc-(nxy/2+1)+nxyW/2);

    %% do Fitts correlation
    deltax = (1./Cx .* sum(sum(Wx.*(yW-s2))))*dxy
    deltay = (1./Cy .* sum(sum(Wy.*(yW-s2))))*dxy;
    offsetSaveX(c)=dx;
    offsetSaveY(c)=dy;
    deltaSaveX(c)=deltax + dxy*(xc-(nxy/2+1)+nxyW/2);
    deltaSaveY(c)=deltay + dxy*(yc-(nxy/2+1)+nxyW/2);

    %% calcualte absolute coordinates of result
    xcenter = deltax + dxy*(xc+nxyW/2+1) - dxy*(nxy/2+1);
    ycenter = deltay + dxy*(yc+nxyW/2+1) - dxy*(nxy/2+1);
    xcenterCENTROID = deltaxCENTROID + dxy*(xc+nxyW/2+1) - dxy*(nxy/2+1);
    ycenterCENTROID = deltayCENTROID + dxy*(yc+nxyW/2+1) - dxy*(nxy/2+1);

    %% display results
    clf;
    subplot(2,2,1);
    imagesc(xW,xW,yW); title('Windowed Input');
    axis image;
    subplot(2,2,2);
```


© Active Optical Systems, LLC

```
    imagesc(xW,xW,s2); title('Windowed Ideal Signal');
    axis image;
    subplot(2,2,3);
    imagesc(xW,xW,yWt); title('Windowed Thresholded Signal for Centroiding');
    axis image;
    subplot(2,2,4);
    imagesc(x,x,y); axis image;
    hold on;
    plot(xcenter,ycenter,'w*','MarkerSize',10);
    plot(dxy*(xc + [0 0 nxyW nxyW 0])-(dxy*(nxy/2-1)),...
        dxy*(yc + [0 nxyW nxyW 0 0])-(dxy*(nxy/2-1)),'w--','LineWidth',3);
    pause(0.05); drawnow;
    if (ppt)
        ToPPT(gcf,'Data',[],1);
    end;

    %% move window if necessary
    if (abs(deltax./dxy)>0.5)
        xc = xc + round(deltax./dxy);
    end;
    if (abs(deltay./dxy)>0.5)
        yc = yc + round(deltay./dxy);
    end;

end;
%% summary
nf;
plot(dxv./dxy,offsetSaveX./dxy,'r--');
hold on;
plot(dxv./dxy,offsetSaveY./dxy,'b--');
plot(dxv./dxy,deltaSaveX./dxy,'r*');
plot(dxv./dxy,deltaSaveY./dxy,'bo');
xlabel('Signal Offset');
ylabel('Offset');
legend('Input X','Input Y','Measured X','Measured Y','Location','Best');
if (ppt) ToPPT(); end;

nf;
plot(dxv./dxy,offsetSaveX./dxy - deltaSaveX./dxy,'r--');
hold on;
plot(dxv./dxy,offsetSaveY./dxy - deltaSaveY./dxy,'b--');
xlabel('Signal Offset');
ylabel('Error');
legend(sprintf('X: RMS=%.2e',RMS(offsetSaveX./dxy - deltaSaveX./dxy)),...
       sprintf('Y: RMS=%.2e',RMS(offsetSaveY./dxy - deltaSaveY./dxy)),...
       'Location','Best');
title('Fitts');
if (ppt) ToPPT(); end;

nf;
plot(dxv./dxy,offsetSaveX./dxy - deltaSaveXCENTROID./dxy,'r--');
hold on;
plot(dxv./dxy,offsetSaveY./dxy - deltaSaveYCENTROID./dxy,'b--');
xlabel('Signal Offset');
ylabel('Error');
```


© Active Optical Systems, LLC

```
legend(sprintf('X: RMS=%.2e',RMS(offsetSaveX./dxy -
deltaSaveXCENTROID./dxy)),...
       sprintf('Y: RMS=%.2e',RMS(offsetSaveY./dxy -
deltaSaveYCENTROID./dxy)),...
       'Location','Best');
title('Centroid');
if (ppt) ToPPT('CENTROID'); end;
```

### 3.5 Evolving Mask 2D Fitts Correlation

```
% 2D Fitts with Mask Evolution
Setup;
ppt=0;

Dap = 1.0;

%full coordinates
nxy = 128;
NoiseMag = 0.0003; % 0.1
w = Dap/80;
dxy = Dap/nxy;
x = (-nxy/2:1:nxy/2-1)*dxy;
[xx,yy] = meshgrid(x,x);

%window coordinates
nxyW = 32;
xW = (-nxyW/2:1:nxyW/2-1)*dxy;
[xxW,yyW] = meshgrid(xW,xW);

nf;
c=0;
dy=0;
%dxv = [zeros(1,10) 0:dxy/4:15*dxy 15*dxy:-dxy/4:0];
NSTEPS = 4;
dxv = [0:dxy/NSTEPS:15*dxy dxy/3/NSTEPS+(15*dxy:-dxy/NSTEPS:0)];
deltax = 0; deltay = 0;
xc = nxy/2+1 - nxyW/2 + deltax;
yc = nxy/2+1 - nxyW/2 + deltay;
started=true;

%THIS IS KEY: The weight needs to be low (<0.1) to
% get a good accurate Fitts estimation
% s2 = (weight)*yW + (1-weight)*s2;
weight = 0.005; % this is the amount of the NEW signal that gets added

for dx = dxv;
    %% setup the noisy measured signal
    noise = NoiseMag.*randn(nxy,nxy);
    dx;
    dy=dx;
    c=c+1;
    s = exp(-1 * ( ((xx-dx)./w).^2 + ((yy-dy)./w).^2) );
    y = s + noise;
    yW = y(xc-1+(1:nxyW),yc-1+(1:nxyW));
```


© Active Optical Systems, LLC

```
%% do thresholded centroiding
thr = 0.2;
yWt = yW - thr;
yWt = yWt .* (yWt>0);
deltaxCENTROID = sum(sum(yWt.*yyW))./sum(sum(yWt));
deltayCENTROID = sum(sum(yWt.*xxW))./sum(sum(yWt));
deltaSaveXCENTROID(c)=deltaxCENTROID + dxy*(xc-(nxy/2+1)+nxyW/2);
deltaSaveYCENTROID(c)=deltayCENTROID + dxy*(yc-(nxy/2+1)+nxyW/2);

%% setup the mask
if (started)
    s2 = yW;
    started = false;
end;
%s2 = exp(-1 * ( (xxW./w).^2 + (yyW./w).^2));
Wy = -gradient(s2);
Wx = -gradient(s2')';
Cx = sum(sum(Wx.^2));
Cy = sum(sum(Wy.^2));
if (dx==min(dxv))
    nf;
    subplot(1,2,1); show(Wx); title('Wx');
    subplot(1,2,2); show(Wy); title('Wy');
    if (ppt)
        ToPPT('Derivates');
    end;
end;

%% do Fitts correlation
deltax = (1./Cx .* sum(sum(Wx.*(yW-s2))))*dxy;
deltay = (1./Cy .* sum(sum(Wy.*(yW-s2))))*dxy;
offsetSaveX(c)=dx;
offsetSaveY(c)=dy;
deltaSaveX(c)=deltax + dxy*(xc-(nxy/2+1)+nxyW/2);
deltaSaveY(c)=deltay + dxy*(yc-(nxy/2+1)+nxyW/2);
disp(sprintf('input:%.2f  Offset: %.2f',dx./dxy,deltax./dxy));

%% calcualte absolute coordinates of result
xcenter = deltax + dxy*(xc+nxyW/2+1) - dxy*(nxy/2+1);
ycenter = deltay + dxy*(yc+nxyW/2+1) - dxy*(nxy/2+1);
xcenterCENTROID = deltaxCENTROID + dxy*(xc+nxyW/2+1) - dxy*(nxy/2+1);
ycenterCENTROID = deltayCENTROID + dxy*(yc+nxyW/2+1) - dxy*(nxy/2+1);

%% display results
clf;
subplot(2,2,1);
imagesc(xW,xW,yW); title('Windowed Input');
axis image;
subplot(2,2,2);
imagesc(xW,xW,s2); title('Windowed Ideal Signal');
axis image;
subplot(2,2,3);
imagesc(xW,xW,yWt); title('Windowed Thresholded Signal for Centroiding');
axis image;
subplot(2,2,4);
```


© Active Optical Systems, LLC

```
    imagesc(x,x,y); axis image;
    hold on;
    plot(xcenter,ycenter,'w*','MarkerSize',10);
    plot(dxy*(xc + [0 0 nxyW nxyW 0])-(dxy*(nxy/2-1)),...
        dxy*(yc + [0 nxyW nxyW 0 0])-(dxy*(nxy/2-1)),'w--','LineWidth',3);
    pause(0.05); drawnow;
    if (ppt)
        ToPPT(gcf,'Data',[],1);
    end;

    %% move window if necessary
    if (abs(deltax./dxy)>0.5)
        xc = xc + round(deltax./dxy);
    end;
    if (abs(deltay./dxy)>0.5)
        yc = yc + round(deltay./dxy);
    end;

    %compute mask
    yW = y(xc-1+(1:nxyW),yc-1+(1:nxyW));
    s2 = (weight)*yW + (1-weight)*s2; %MUCH better than before

end;
%% summary
nf;
plot(dxv./dxy,offsetSaveX./dxy,'r--');
hold on;
plot(dxv./dxy,offsetSaveY./dxy,'b--');
plot(dxv./dxy,deltaSaveX./dxy,'r*');
plot(dxv./dxy,deltaSaveY./dxy,'bo');
xlabel('Signal Offset');
ylabel('Offset');
legend('Input X','Input Y','Measured X','Measured Y','Location','Best');
if (ppt) ToPPT(); end;

nf;
plot(dxv./dxy,offsetSaveX./dxy - deltaSaveX./dxy,'r--');
hold on;
plot(dxv./dxy,offsetSaveY./dxy - deltaSaveY./dxy,'b--');
xlabel('Signal Offset(pixels)');
ylabel('Error(pixels)');
legend(sprintf('X: RMS=%.2e',RMS(offsetSaveX./dxy - deltaSaveX./dxy)),...
       sprintf('Y: RMS=%.2e',RMS(offsetSaveY./dxy - deltaSaveY./dxy)),...
       'Location','Best');
title('Fitts');
if (ppt) ToPPT(); end;

nf;
plot(dxv./dxy,offsetSaveX./dxy - deltaSaveXCENTROID./dxy,'r--');
hold on;
plot(dxv./dxy,offsetSaveY./dxy - deltaSaveYCENTROID./dxy,'b--');
xlabel('Signal Offset (pixels)');
ylabel('Error (pixels)');
legend(sprintf('X: RMS=%.2e',RMS(offsetSaveX./dxy -
deltaSaveXCENTROID./dxy)),...
```


© Active Optical Systems, LLC

```
       sprintf('Y: RMS=%.2e',RMS(offsetSaveY./dxy -
deltaSaveYCENTROID./dxy)),...
       'Location','Best');
title('Centroid');
if (ppt) ToPPT('CENTROID'); end;
```

### 3.6 2D Fitts Correlation – Complex Signal – Small Window – Evolving Mask

```
% 2D Fitts with Mask Evolution - Complex Signal
Setup;
ppt=1;

%full coordinates
Dap = 1.0;
nxy = 128;

signalType=1; %Gaussian
w = Dap/40; %size of the Gaussian
ppttitle='Gaussian';

signalType=2; %Sinusoid
w = Dap/4; %size of the sine
ppttitle='Sinusoid in two Axes';

signalType=3; %Sinusoid 2
w = Dap/6; %size of the sinusoid
ppttitle='Product of Sinusoidal Signals in Two Axes';

if (ppt)
    TitleSlidePowerPoint(ppttitle);
end;

NoiseMag = 0.05; % 0.1
%THIS IS KEY: The weight needs to be low (<0.1) to
% get a good accurate Fitts estimation
% s2 = (weight)*yW + (1-weight)*s2;
weight = 0.05; % this is the amount of the NEW signal that gets added
nxyW = 32; %size of the window
NSTEPS = 4; %number of offset steps per pixel
CentroidThreshold = 0.2; %centroid threshold

if (ppt)
    vvv = whos;
    for ii=1:length(vvv);
        txt{ii}=sprintf('%s = %f',vvv(ii).name,eval(vvv(ii).name));
    end;
    TextToPowerPoint(txt,'Variables',18,20);
end;

dxy = Dap/nxy;
x = (-nxy/2:1:nxy/2-1)*dxy;
[xx,yy] = meshgrid(x,x);
```


© Active Optical Systems, LLC

```
%window coordinates
xW = (-nxyW/2:1:nxyW/2-1)*dxy;
[xxW,yyW] = meshgrid(xW,xW);

nf;
c=0;
dy=0;
%dxv = [zeros(1,10) 0:dxy/4:15*dxy 15*dxy:-dxy/4:0];
dxv = [0:dxy/NSTEPS:15*dxy dxy/3/NSTEPS+(15*dxy:-dxy/NSTEPS:0)];
deltax = 0; deltay = 0;
xc = nxy/2+1 - nxyW/2 + deltax;
yc = nxy/2+1 - nxyW/2 + deltay;
started=true; %flag for first iteration indication

nf; plot(dxv./dxy,'b*'); xlabel('Step'); ylabel('Input Signal Offset');
if (ppt) ToPPT('Input Offset'); end;

for dx = dxv;
    %% setup the noisy measured signal
    noise = NoiseMag.*randn(nxy,nxy);
    dx;
    dy=dx;
    c=c+1;
    if (signalType==1)
        s = exp(-1 * ( ((xx-dx)./w).^2 + ((yy-dy)./w).^2) );
    elseif (signalType==2)
        s = (cos(2.0*pi.*(((xx-dx)./w) + ((yy-dy)./w)))+1)/2;
    else
        s = ((cos(2.0*pi.*(((xx-dx)./w)))+1)/2) .* ((cos(2.0*pi.*(((yy-
dy)./(w/2.0))))+1)/2);
    end;
    y = s + noise;
    yW = y(xc-1+(1:nxyW),yc-1+(1:nxyW));

    %% do thresholded centroiding
    yWt = yW - CentroidThreshold;
    yWt = yWt .* (yWt>0);
    deltaxCENTROID = sum(sum(yWt.*yyW))./sum(sum(yWt));
    deltayCENTROID = sum(sum(yWt.*xxW))./sum(sum(yWt));
    deltaSaveXCENTROID(c) = deltaxCENTROID + dxy*(xc-(nxy/2+1)+nxyW/2);
    deltaSaveYCENTROID(c) = deltayCENTROID + dxy*(yc-(nxy/2+1)+nxyW/2);

    %% setup the mask
    if (started)
        s2 = yW;
        started = false;
    end;
    %s2 = exp(-1 * ( (xxW./w).^2 + (yyW./w).^2));
    Wy = -gradient(s2);
    Wx = -gradient(s2')';
    Cx = sum(sum(Wx.^2));
    Cy = sum(sum(Wy.^2));
    if (dx==min(dxv))
        nf;
        subplot(1,2,1); show(Wx); title('Wx');
        subplot(1,2,2); show(Wy); title('Wy');
```


© Active Optical Systems, LLC

```
        if (ppt)
            ToPPT('Derivates');
        end;
    end;

    %% do Fitts correlation
    deltax = (1./Cx .* sum(sum(Wx.*(yW-s2))))*dxy;
    deltay = (1./Cy .* sum(sum(Wy.*(yW-s2))))*dxy;
    offsetSaveX(c)=dx;
    offsetSaveY(c)=dy;
    deltaSaveX(c)=deltax + dxy*(xc-(nxy/2+1)+nxyW/2);
    deltaSaveY(c)=deltay + dxy*(yc-(nxy/2+1)+nxyW/2);
    disp(sprintf('input:%.2f  Offset: %.2f',dx./dxy,deltax./dxy));

    %% calcualte absolute coordinates of result
    xcenter = deltax + dxy*(xc+nxyW/2+1) - dxy*(nxy/2+1);
    ycenter = deltay + dxy*(yc+nxyW/2+1) - dxy*(nxy/2+1);
    xcenterCENTROID = deltaxCENTROID + dxy*(xc+nxyW/2+1) - dxy*(nxy/2+1);
    ycenterCENTROID = deltayCENTROID + dxy*(yc+nxyW/2+1) - dxy*(nxy/2+1);

    %% display results
    clf;
    subplot(2,2,1);
    imagesc(xW,xW,yW); title('Windowed Input');
    axis image;
    subplot(2,2,2);
    imagesc(xW,xW,s2); title('Windowed Ideal Signal');
    axis image;
    subplot(2,2,3);
    imagesc(xW,xW,yWt); title('Windowed Thresholded Signal for Centroiding');
    axis image;
    subplot(2,2,4);
    imagesc(x,x,y); axis image;
    hold on;
    plot(xcenter,ycenter,'w*','MarkerSize',10);
    plot(dxy*(xc + [0 0 nxyW nxyW 0])-(dxy*(nxy/2-1)),...
        dxy*(yc + [0 nxyW nxyW 0 0])-(dxy*(nxy/2-1)),'w--','LineWidth',3);
    pause(0.05); drawnow;
    if (ppt)
        ToPPT(gcf,'Data',[],1);
    end;

    %% move window if necessary
    if (abs(deltax./dxy)>0.5)
        xc = xc + round(deltax./dxy);
    end;
    if (abs(deltay./dxy)>0.5)
        yc = yc + round(deltay./dxy);
    end;

    % compute mask
    yW = y(xc-1+(1:nxyW),yc-1+(1:nxyW));
    s2 = (weight)*yW + (1-weight)*s2; %MUCH better than before

end;
%% summary
```


© Active Optical Systems, LLC

```
nf;
plot(dxv./dxy,offsetSaveX./dxy,'r--');
hold on;
plot(dxv./dxy,offsetSaveY./dxy,'b--');
plot(dxv./dxy,deltaSaveX./dxy,'r*');
plot(dxv./dxy,deltaSaveY./dxy,'bo');
xlabel('Signal Offset');
ylabel('Offset');
legend('Input X','Input Y','Measured X','Measured Y','Location','Best');
if (ppt) ToPPT(); end;

nf;
plot(dxv./dxy,offsetSaveX./dxy - deltaSaveX./dxy,'r--');
hold on;
plot(dxv./dxy,offsetSaveY./dxy - deltaSaveY./dxy,'b--');
xlabel('Signal Offset(pixels)');
ylabel('Error(pixels)');
legend(sprintf('X: RMS=%.2e',RMS(offsetSaveX./dxy - deltaSaveX./dxy)),...
       sprintf('Y: RMS=%.2e',RMS(offsetSaveY./dxy - deltaSaveY./dxy)),...
       'Location','Best');
title('Fitts');
if (ppt) ToPPT(); end;

nf;
plot(dxv./dxy,offsetSaveX./dxy - deltaSaveXCENTROID./dxy,'r--');
hold on;
plot(dxv./dxy,offsetSaveY./dxy - deltaSaveYCENTROID./dxy,'b--');
xlabel('Signal Offset (pixels)');
ylabel('Error (pixels)');
legend(sprintf('X: RMS=%.2e',RMS(offsetSaveX./dxy -
deltaSaveXCENTROID./dxy)),...
       sprintf('Y: RMS=%.2e',RMS(offsetSaveY./dxy -
deltaSaveYCENTROID./dxy)),...
       'Location','Best');
title('Centroid');
if (ppt) ToPPT('CENTROID'); end;
```


© Active Optical Systems, LLC