Object Tracking using Image Registration and Kalman Filter

VPS Naidu and J.R. Raol
Multi Sensor Data Fusion Lab
Flight Mechanics and Control Division
National Aerospace Laboratories
Bangalore-17, India
Email: vpsnaidu@gmail.com

***Abstract*-Image registration and Kalman filter based object tracking algorithm is presented. Image registration algorithms viz., sum absolute difference (SAD) and normalized cross correlation (NCC) algorithms are used to find the centroid of the object of interest and Kalman filter is used to track the centroid of the target. Paraboloid interpolation has been used to compute the centroid in sub pixel accuracy. It was observed that in presence of salt and pepper noise, the SAD algorithm performed better and in presence of Gaussian noise, the NCC performed better.**

I. INTRODUCTION

Many real world applications, military as well as civilian, require accurate tracking of moving targets acquired by imaging sensors. In military applications, tracking may be used in reconnaissance such as that from a satellite where continually updated knowledge of a target's position may be useful. In civilian applications, target tracking can be of much use in autonomous vehicles, home security etc. Accurate target tracking can be used in many instances to alleviate the need for constant human intervention and thus may help to achieve a much higher degree of autonomy and dependability.

The general procedure for tracking using data from imaging sensors is as follows. The target to be tracked is first specified by a human operator. An image registration algorithm then searches for the target in each subsequent image obtained by the imaging sensor. The measurement resulting from the image registration algorithm is passed to a target state estimator. The estimator continuously estimates the position of the target based on the measurements it has received at any point in time. The advantage of using an estimator is that along with the position estimates, it gives an estimate of the accuracy of the estimation. It takes care of noisy or missing measurements and continuously provides the best estimate depending on the available measurements, given the measurement/sensor accuracy. In this paper, two of the image registration algorithms viz., Sum Absolute Difference (SAD) and Normalized Cross Correlation (NCC) are used to find the centroid of the object of interest. Kalman filter is used to track the centroid to maintain the tract. The proposed algorithms are implemented and validated using simulated data.

II. IMAGE REGISTRATION ALGORITHMS

An image registration algorithm is used to find the centroid of the target of interest in current frame by registering the target reference image with current image frame. One of the following two algorithms are generally used for this purpose

A. *Sum of Absolute Differences*

The sum absolute differences of two 1-D discrete signals $I_c(x)$ of length M and $I_r(x)$ of length P is calculated using the formula

$$SAD(x) = \sum_{i=0}^{P-1} \left| I_c(x+i) - I_r(i) \right|, \quad x = 0,1,2,...,M-1 \tag{1}$$

In this method, the reference signal is aligned with each pixel in the search/current frame and then subtracted from it. This yields another signal where each pixel contains the sum of the absolute value of the differences between reference signal and the search frame, had the reference signal been aligned at that pixel in the search frame. The sum of the absolute differences will be minimum at the position at which similarity is maximum. The sum absolute difference of two 2-dimensional images $I_c(x, y)$ of length $M\text{x}N$ and $I_r(x, y)$ of length $P\text{x}Q$ is calculated using the formula [1]

$$SAD(x, y) = \sum_{i=0}^{P-1}\sum_{j=0}^{Q-1} \left| I_c(x+i, y+j) - I_r(i, j) \right|, \quad \begin{matrix} x = 0,1,2,...,M-1 \\ y = 0,1,2,...,N-1 \end{matrix} \tag{2}$$

B. *Normalized Cross Correlation*

Cross-correlation is a measure of the similarity of two signals or images. The cross correlation of two 2-D discrete signals $I_c(x, y)$ and $I_r(x, y)$ of dimensions $M\text{x}N$ and $P\text{x}Q$ yields a 2-D correlation sequence of dimensions $(M+P-1)\text{x}(N+Q-1)$ and is calculated using the formula

$$CC(x,y) = \sum_{i=0}^{N-1}\sum_{j=0}^{M-1} I_c(x+i, y+j) I_r(i, j) \quad \begin{matrix} x = 0,1,2,...,M+P-1 \\ y = 0,1,2,...,N+Q-1 \end{matrix} \tag{3}$$

The normalized cross correlation of two 2-D discrete signals $I_c(x, y)$ and $I_r(x, y)$ of dimensions $M\text{x}N$ and $P\text{x}Q$ yields a 2-D correlation sequence of dimensions

$(M+P-1)\text{x}(N+Q-1)$ and is calculated using the formula [2,3]

$$NCC(x,y)=\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}\frac{I_c(x+i,y+j)I_r(i,j)}{\sqrt{\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}I_c^2(x+i,y+j)}\sqrt{\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}I_r^2(i,j)}} \quad (4)$$

One can see that the numerator in eq. (4) is a convolution of the reference image and current frame. For a current frame of size M^2 and a reference image of size P^2, it requires approximately $P^2(M-P+1)^2$ additions and same number of multiplications. Cross correlation can be computed by:

$$F^{-1}\left\{F(I_c)F^*(I_r)\right\} \quad (5)$$

where F is the Fourier transform and superscript star indicates complex conjugate

The complexity using FFT is $12M^2\log_2 M$ real multiplications and $18M^2\log_2 M$ real additions/subtractions [7]. If P approaches M or larger M & P then transform method becomes faster otherwise the direct convolution becomes faster.

The above algorithms provide the point in the current frame around which similarity to the reference image is maximum. In other words these algorithms return the position of the centroid of the reference image inside the current frame.

III. INTERPOLATION

An image may often correspond to a large physical area, and for better accuracy in tracking, interpolation may be used. Interpolation may be of many types: linear, polynomial, spline etc. The centroid found by the centroid finding algorithms has integral or half integer values. These values are interpolated using the function for a paraboloid, based on the following formulae to achieve sub-pixel accuracy [1]:

$$\begin{aligned} a &= q(x_0,y_0) \\ b &= \frac{1}{2}(q(x_1,y_0)-q(x_{-1},y_0)) \\ c &= \frac{1}{2}(q(x_0,y_1)-q(x_0,y_{-1})) \\ d &= -q(x_0,y_0)+\frac{1}{2}q(x_1,y_0)+\frac{1}{2}q(x_{-1},y_0) \\ e &= -q(x_0,y_0)+\frac{1}{2}q(x_0,y_1)+\frac{1}{2}q(x_0,y_{-1}) \end{aligned} \quad (6)$$

Actual centroid of the target is $(x_c,y_c)=\left(\frac{b}{2d},\frac{c}{2e}\right)$ (7)

where, q is a 3x3 matrix with the peak at the center and the immediate neighbors of the peak at their corresponding positions.

IV. NOISE GENERATION AND SPATIAL FILTERING

A. Noise Generation

Salt & Pepper Noise: This type of noise may be caused by the errors in image data transmission, malfunctioning pixel elements in the camera sensors, faulty memory locations, or timing errors in the digitization process. The corrupted pixels are set to zero or maximum value, which gives the image a salt and pepper like appearance. Uncorrupted pixels remain unchanged.

$$x(i,j)=\begin{cases} 0 & rand<0.5d \\ 255 & 0.5d<rand<d \\ s(i,j) & rand\geq d \end{cases} \quad (8)$$

where, $rand$ is a uniform distribution of random numbers in the interval zero to one, d is a positive real number denoting the noise density, $s(i,j)$ is the original/true image pixel and $x(i,j)$ is the noisy image pixel

Gaussian Noise: This type of noise is due to electronic noise in the image acquisition system. The noise can be generated with zero mean Gaussian distribution described by its standard deviation (σ).

$$\begin{aligned} v(i,j) &= randn*\sigma \\ x(i,j) &= s(i,j)+v(i,j) \end{aligned} \quad (9)$$

where $randn$ is normal distribution of random numbers with zero mean and unit standard deviation and σ is standard deviation

B. Spatial Filters

Two of the most common spatial filters are used for handling noisy image data are:

Mean Filter: This is the simplest linear spatial filter and is sometimes called average, smooth, box or uniform filter. It is an intuitive and easily implemented method for reducing noise in an image. It reduces the amount of intensity variation between one pixel and its neighbours. The principle of mean filtering is very simple. It is a simple sliding window spatial filter that replaces the centre pixel value in the window with the average (mean) of all pixel values in the window. This approach has the effect of purging the pixel values which are unrepresentative of their neighbors. A mean filter is generally implemented by convolution i.e. computing the convolution of the noisy image with a kernel. The kernel represents the shape and size of the neighborhood to be sampled when calculating the mean. The coefficients in the kernel (convolution mask) are non-negative and equal. Masks of different sizes can be obtained as:

$$h_{mk}=\frac{ones(k,k)}{k^2} \quad (10)$$

where, $ones(k,k)$ is a k x k square matrix having all elements as unity, k indicates the mask size and h_{mk} is the convolution mask

The filter is normalized so that $\sum h_{mk}(i,j)=1$ which ensures that the resulting image has the same contrast as the input image.

Median Filter: This filter is also called rank filter. It is a non-linear spatial filter that is good at removing impulse noise. This filter often does a better job than the mean filter of preserving useful detail in the image. The median filtering operation is

performed on an image by applying the sliding window concept. The median is calculated by first sorting all the pixel values from the surrounding neighborhood and then replacing the pixel being considered with the middle pixel value. Unrepresentative pixels in a neighborhood will not affect the median value significantly. This filter would not create any unrealistic pixel values when the filter straddles an edge because the median value is exactly equal to one of the pixel value in the neighborhood.

It has been found that the median filter performs better than the mean filter in the presence of salt & pepper noise whereas a mean filter performs better than the median filter in the presence of Gaussian noise [4,5].

V. TRAKING ALGORITHM

The centroid found using the image registration algorithms is fed into a simple Kalman filter target tracker [6,7]. In simple case the target moves in a straight line with constant velocity. In reality though, the velocity of a target is rarely constant, and in order to allow for this, a noise component called the state noise is included in the model. If the target position and velocity at time k are given by the state vector $X(k)$, then under the constant velocity assumption the state at time $k+1$ will be given by: $X(k+1) = \Phi X(k) + w(k)$

$$\Rightarrow \begin{pmatrix} x(k+1) \\ \dot{x}(k+1) \\ y(k+1) \\ \dot{y}(k+1) \end{pmatrix} = \begin{pmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x(k) \\ \dot{x}(k) \\ y(k) \\ \dot{y}(k) \end{pmatrix} + \begin{pmatrix} w_x(k) \\ \dot{w}_x(k) \\ w_y(k) \\ \dot{w}_y(k) \end{pmatrix} \quad (11)$$

where, Φ is the state transition matrix, w is the additive noise component, which is assumed to be having a normal distribution.

The target model noise covariance matrix $Q = E\{ww^T\}$ is assumed to be known, where $E\{\}$ denotes statistical expectation. If $z(k)$ denotes the measurement vector of the target, then it is assumed that $z(k) = HX(k) + v(k)$, where v is the measurement noise, assumed to be independent of the state noise $w(k)$ and normally distributed. If only the position is measured then the equation becomes:

$$\begin{pmatrix} z_x(k) \\ z_y(k) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x(k) \\ \dot{x}(k) \\ y(k) \\ \dot{y}(k) \end{pmatrix} + \begin{pmatrix} v_x(k) \\ v_y(k) \end{pmatrix} \quad (12)$$

The measurement noise v is assumed to have known covariance $R = E\{vv^T\}$.

The purpose of the Kalman filter is to estimate the true state vector (the position and velocity) of the target based on the measurements it has received so far at any point in time. On receiving a new measurement, the Kalman filter updates the previous estimate based on the new information contained in the measurement. This is done by calculating the error in its prediction, which is called the innovation, so named because of the new information obtained from the new measurement that it represents. The complete set of equations describing the Kalman filter is:

$$\tilde{X}(k+1 \mid k) = \Phi \hat{X}(k \mid k) \quad \text{Prediction} \quad (13)$$

$$\vartheta = z(k+1) - H\tilde{X}(k+1 \mid k) \quad \text{Innovation} \quad (14)$$

$$\hat{X}(k+1 \mid k+1) = \tilde{X}(k+1 \mid k) + K\vartheta \quad \text{Estimate} \quad (15)$$

$$K = \tilde{P}(k+1 \mid k) H^T S^{-1} \quad \text{Kalman gain} \quad (16)$$

$$S = H\tilde{P}(k+1 \mid k) H^T + R \quad \text{Innovation covariance} \quad (17)$$

$$\hat{P}(k+1 \mid k+1) = (I - KH)\tilde{P}(k+1 \mid k) \quad \text{Estimate covariance} \quad (18)$$

$$\tilde{P}(k+1 \mid k) = \Phi \hat{P}(k \mid k) \Phi^T + Q \quad \text{Prediction covariance} \quad (19)$$

where, $\hat{X}(k \mid k)$ is the estimate at time k after taking the measurement into account and $\tilde{X}(k+1 \mid k)$ is the Kalman filter prediction of the state vector before measurement at time k.

VI. DATA SIMULATION

The data simulator is implemented in PC MATLAB. The Graphical User Interface (GUI) of the simulator is shown in Fig-1. The target may be simulated as a rectangular block having a Gaussian distribution of intensities around its centre. A Gaussian distribution is given by the equation:

$$f_g(x) = \frac{1}{\sqrt{2\prod\sigma^2}} e^{-\frac{(x-a)^2}{2\sigma^2}} \quad (20)$$

where, σ is the standard deviation and a is the location parameter or mean which in this case corresponds to the center of the rectangle

VII. RESULTS AND DISCUSSIONS

Three simulated data sets are used to test the proposed image registration algorithms for target tracking along with the Kalman filter. The graphical user interface for target tracking is shown in Fig-2. The tracking performance is evaluated using performance check metrics [8].

i. The percentage fit error (PFE) in x & y positions:

$$\text{PFE } x = 100 * \frac{norm(x_t - \hat{x})}{norm(x_t)} \quad (21)$$

$$\text{PFE } y = 100 * \frac{norm(y_t - \hat{y})}{norm(y_t)} \quad (22)$$

where x_t is the true x-position and $\hat{x}$ is the estimated x-position

ii. Root mean square error in position:

$$\text{RMSPE} = \sqrt{\frac{1}{M}\sum_{i=1}^{M} \frac{(x_t(i) - \hat{x}(i))^2 + (y_t(i) - \hat{y}(i))^2}{2}} \quad (23)$$

iii. Mean absolute error in x & y positions

$$\text{MAEx} = \frac{1}{M}\sum_{i=1}^{M}\left|x_t(i) - \hat{x}(i)\right| \quad (24)$$

$$\text{MAEy} = \frac{1}{M}\sum_{i=1}^{M}\left|y_t(i) - \hat{y}(i)\right| \quad (25)$$

Data Set1: 100 frames of noiseless video in which the target moves with a constant velocity from the bottom left corner to the top right corner processed without any filters or interpolation. The performance check metrics viz. PFE, RMSPE and MAE are shown in Table-1. All these metrics are zero, since there is no noise in the data. Hence, the filter performed very well.

Data Set2: 100 frames of video corrupted by salt & pepper noise of noise density 0.05 in which the target moves with constant acceleration in a parabolic trajectory from the bottom left corner to the bottom right corner. The PFE, RMSPE and MAE are shown in Fig-3. It is observed that interpolation reduces the estimation error in states when the images are corrupted with salt & pepper noise.

Data Set 3: 100 frames of video corrupted by Gaussian noise of variance 0.04 in which the target moves with a constant velocity from the top right corner to the bottom right corner. The performance checks are shown in Fig-4. In the case of image data corrupted with Gaussian noise, interpolation reduces the error when the input data is not treated with the mean filter. However, on being treated with the mean filter, interpolation causes the error to increase. This may be due to the blurring effect of the mean filter, which causes the values to be more uniform, thus throwing the interpolation algorithm off-track.

It is observed from these results that spatial filter (mean filter in this case) improves the tracking performance as does interpolation though only marginally. It is also observed that NCC fares better than SAD in the estimation of the state vector in this data. When comparing SAD and NCC as image registration techniques, it is observed from the Fig-3&4 that the performance of SAD is better than that of NCC when the input image data is corrupted by salt & pepper noise. However, when the input image data is corrupted by Gaussian noise, the performance of NCC is better than that of SAD.

TABLE I
PFE, RMSPE AND MASE – DATA SET 1

	SAD	NCC
PFEx	0	0
PFEy	0	0
RMSPE	0	0
MAEx	0	0
MaeY	0	0

PFEx: percentage fit error in x-position, PFEy: percentage fit error in y-position, RMSPE: root mean square error in positon, MAEx: mean absolute error in x-position and MAEy: mean absolute error in y-position

VIII. CONCLUSIONS

Image registration algorithms viz. SAD & NCC, spatial filtering algorithms as a preprocessing step and an interpolation algorithm to achieve sub-pixel accuracy were implemented in PC MATLAB. Subsequently, a Kalman filter was used to track the centroid of the target obtained using the image registration algorithm. Pertaining to the comparison of SAD and NCC as image registration techniques: a) in the absence of noise, both image registration techniques, proved to be equally accurate, b) in the presence of salt & pepper noise, SAD proved to be more accurate than NCC and c) in the presence of Gaussian noise, NCC proved to be more accurate than SAD. Pertaining to the effect of spatial filtering: a) in the presence of salt & pepper noise, the median filter drastically reduced the error in state estimation when either of the image registration techniques was used and b) in the presence of Gaussian noise, the mean filter drastically reduced the error in state estimation when either of the image registration techniques was used. Pertaining to the effect of interpolation, it reduces the error in the state estimation. It is thus concluded that the choice of image registration technique in any application depends on the characteristics of the input image data that may be expected. However, in all cases, spatial filtering may be used to achieve much better performance. Also, in almost all cases, interpolation too may be used to further improve the performance at little computational cost.

REFERENCES

[1] Michael K. Cheezum, William F. Walker and William H. Guilford, "Quantitative Comparsion of Algorithms for Tracking Single Flourescent Particles", *Biophysical Journal*, Vol.81, pp2378-2388, Oct. 2001.

[2] William K. Pratt, "Correlation Techniques of Image Registration", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-10, No.3, pp 353-358, May 1974.

[3] Shoude Chang, Chander P. Grover, "Centroid detection based on optical correlation", *Optical Engineering*, Vol. 41, No. 10, pp 2479-2486, October 2002.

[4] Rafael C. Gonzalez and Richard E. Woods, "Digital Image Processing", *Addison-Wesley Inc.*, New York, 1993.

[5] Gonzalo R. Arce, "Nonlinear Signal Processing – A statistical approach", *Wiley-Interscience Inc.*, Publication, USA, 2005.

[6] E V Stansfield, "Introduction to Kalman Filters", *Thales Research Ltd,* Reading, Heckfield Place, 7th March 2001.

[7] J.P. Liwis, "Fast Template Matching", *Vision Interface*, pp. 120-123, 1995.

[8] VPS Naidu, G. Girija and JR Raol, "Data Association and Fusion Algorithms for Tracking in presence of Measurement Loss", *Journal of the Institution of Engineers (I)*, Vol. 86, pp.17-28, May 2005.

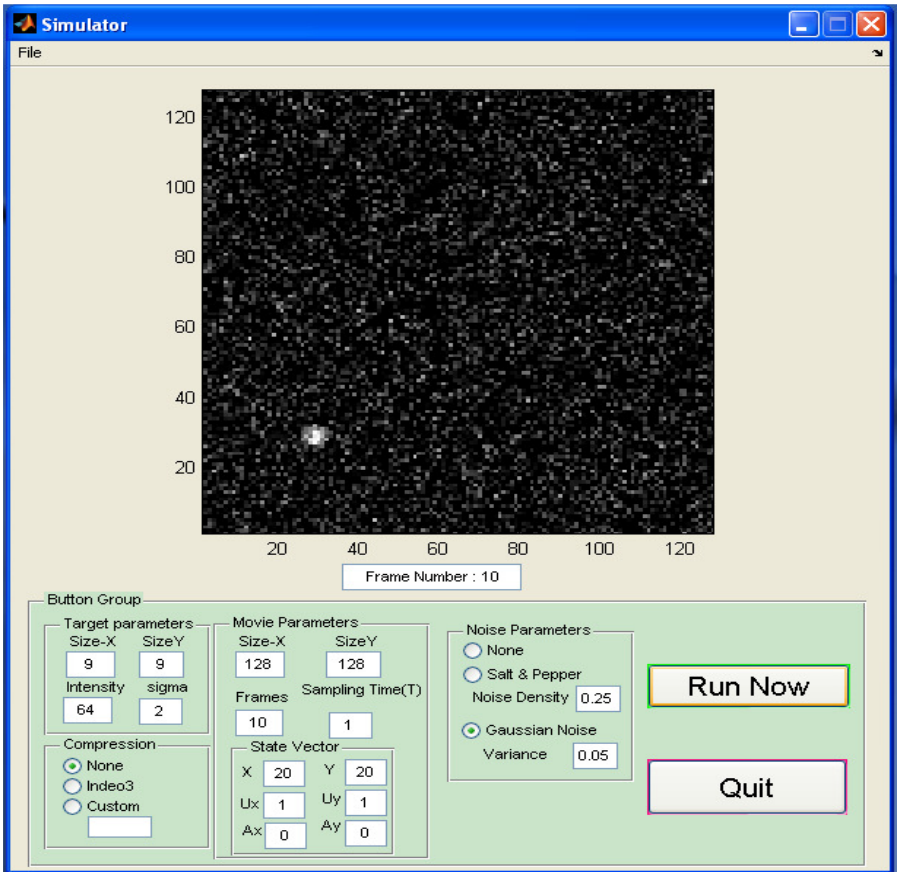

Fig. 1 GUI for scenario simulator

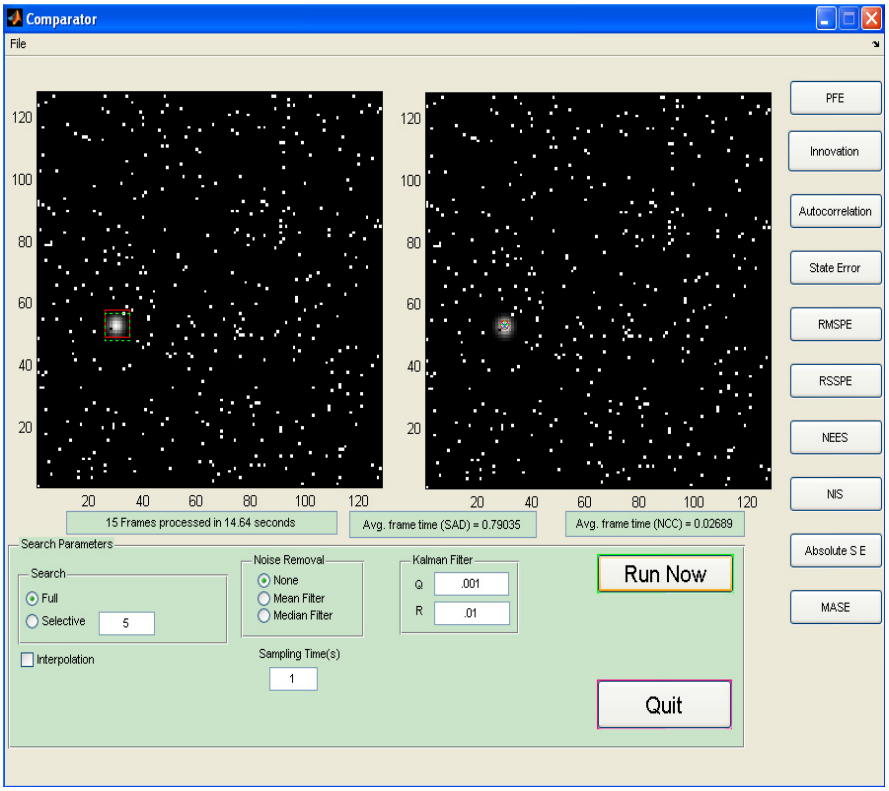

Fig. 2 GUI for object tracker

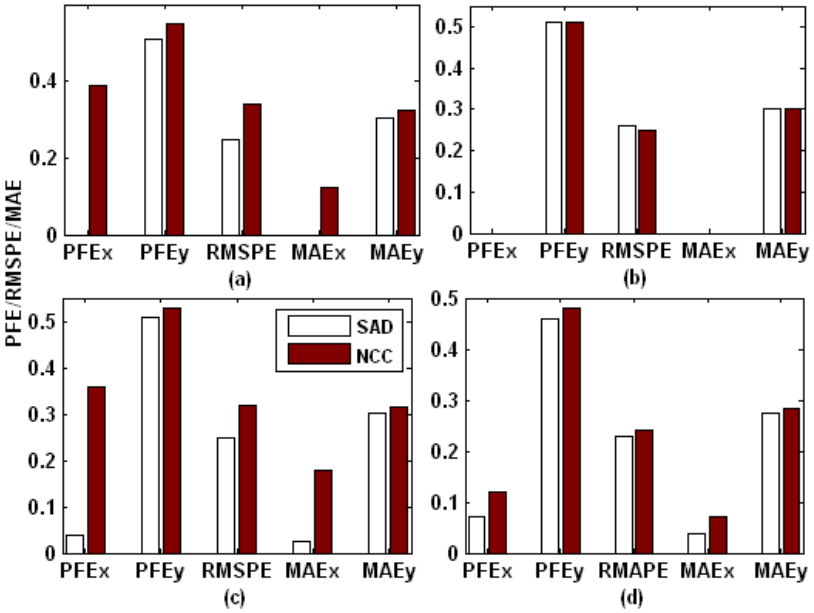

Fig. 3 Performance evaluation metrics (a) without filter and without interpolation, (b) with mean filter and without interpolation, (c) without filter and with interpolation and (d) with mean filter and interpolation – Data Set 2

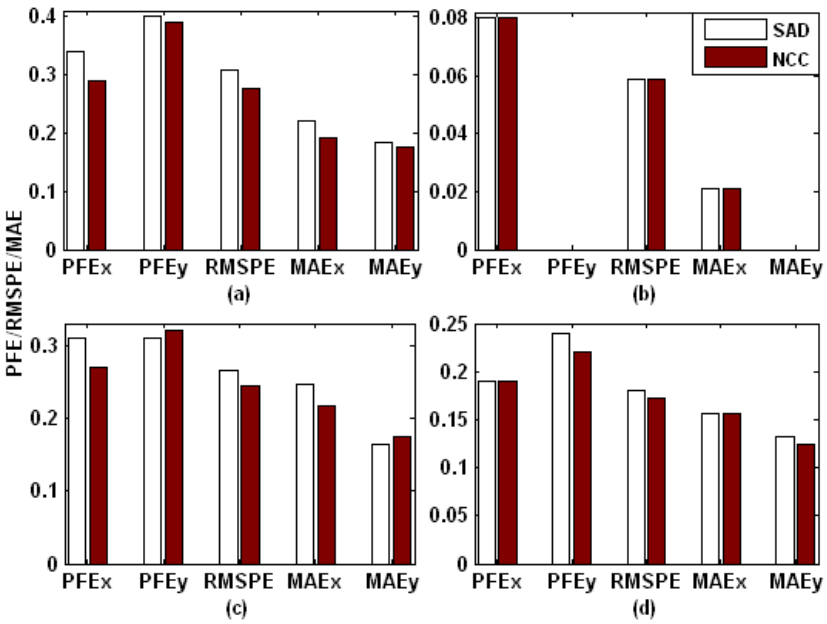

Fig. 4 Performance evaluation metrics (a) without filter and without interpolation, (b) with mean filter and without interpolation, (c) without filter and with interpolation and (d) with mean filter and interpolation – Data Set 3