Contrast Enhancement of Missile Video Sequences via Image Stabilization and Product Correlation

Seung–Jin Choi, Richard R. Schultz, Robert L. Stevenson,

Yih–Fang Huang, and Ruey–Wen Liu

Corresponding Author:
Robert L. Stevenson
Laboratory for Image and Signal Analysis
Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556
Phone: (219) 631–8308
Fax: (219) 631–4393
E-Mail: Stevenson.1@nd.edu

May 9, 1995

Submitted to *Optical Engineering*, October 1994.

This research was presented in part at the *IS&T/SPIE Conference on Image and Video Processing II*, (San Jose, CA), February 1994.

The authors gratefully acknowledge the support of the Air Force Office of Scientific Research, through the auspices of Rome Laboratories, Rome, NY, Contract F30602–92–C–0138. Research and data sets included in this article are unclassified material.

Authors are with the Laboratory for Image and Signal Analysis, Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556. Please address all correspondence to: Robert L. Stevenson, Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556, Phone: (219) 631–8308, Fax: (219) 631–4393, E-Mail: Stevenson.1@nd.edu

Abstract

An image sequence containing a missile in flight often contains a strong plume signature expelled by the missile, with a weak signal corresponding to the missile hardbody. Enhancement of the missile signal is necessary to accurately track the trajectory throughout the sequence. A parametric motion estimation algorithm is proposed for the passive stabilization of the data set. By stabilizing the image sequence with respect to the missile vacuum core, a registered sequence is produced with the missile located in the same position within each frame. Missile contrast can then be enhanced by applying a novel technique known as product correlation to the stabilized data. Product correlation is presented in the context of higher-order statistics, and it offers a computationally efficient means of obtaining sample moments of high orders collectively. Simulations with missile sequences acquired in the infrared and visible portions of the spectrum show that image stabilization followed by product correlation successfully enhances missile signal contrast, particularly in image sequences characterized by low signal-to-noise ratios.

Keywords: Bi-linear time series models, higher-order statistics, image sequence processing, image stabilization, parametric motion estimation, product correlation.

Permission to publish this abstract separately is granted.

I. Introduction

Tracking a moving object such as a missile in flight has been an active research area in the past few decades. If the object is distinguishable from the background, we can easily locate it within a video sequence using standard feature extraction techniques. Quite often, a video sequence contains an object of interest which is obscured and cannot be tracked easily. Enhancement of the object is required so that it can be followed throughout the sequence. Temporal correlation present in a video sequence can be used to enhance the object compared to that of the noisy background. However, in video data, an object is not located at the same position throughout the sequence. Therefore it must be registered in order to compute the correct temporal correlations for the object. In this paper, the object of interest is a missile hardbody. We will develop and study techniques for the stabilization and enhancement of the missile in flight.

Image stabilization performs a transformation on the data within a video sequence containing a missile signal, so that the missile hardbody is in the same position within each frame. The proposed algorithm is *passive* in nature; *i.e.*, the general location of the missile is inferred from the image data only, without aid of illuminating lasers used to pinpoint the missile hardbody. In this research, stabilization was designed primarily for sequences acquired in the infrared spectrum, since the missile vacuum core was selected as the stable feature in the data. Essentially, the algorithm consists of three steps. First, the vacuum core position is estimated by a least squares fit to data in the vicinity of the data peak, providing an initial approximation of the data translation parameters. Next, an estimate of the missile direction is computed using principles of morphological processing. This provides an accurate estimate of the missile orientation. Finally, a Kalman filter is used to smooth the missile vacuum core position estimate in time. Data within each video image undergoes a linear transformation, so that the vacuum core is located at the center of each frame. The result is a video sequence with the missile in the same position and orientation throughout.

Once the video sequence has been registered by image stabilization, the computation of temporal correlation is required for contrast enhancement. In the past, second-order correlations have been used almost exclusively. Higher-order statistics (HOS), represented by higher-order correlations, cumulants, or polyspectra (Fourier transform of cumulants), carry the extra information which is not found in the second-order statistics. For instance, HOS contain phase information, while second-order statistics do not. For a recent review on HOS, the reader is directed to Mendel [Mendel91] and references therein.

The statistical properties of a random variable can be completely characterized by the cumulative distribution function (CDF) or characteristic function (CF). If all moments of a random variable exist, one can obtain the CF via a Taylor series expansion. Specifically, if all moments of a random variable are known, statistical properties of the random variable are completely characterized by the CF. For example, the Gaussian random variable is completely characterized by the first two moments, namely the mean and variance. Although stochastic processes are often assumed to be Gaussian for the sake of simplicity, most actual processes possess moments higher than order two. Thus, HOS are important in the characterization of real world signals.

Major obstacles exist to directly using higher-order moments, primarily due to analytical complexity and computational inefficiency. Estimating higher-order moments using higher-order sample moments requires an excessive number of samples, and therefore an enormous amount of computation. *Product correlation* (PC) offers a computationally efficient means of obtaining sample moments of higher orders collectively. This method is recursive in nature, and it yields collective sample moments of higher orders in one variable. PC was developed to aid in the contrast enhancement of the missile hardbody signal in video sequences exhibiting a low signal-to-noise ratio (SNR). The collective sample correlations produced by PC sufficiently enhance image contrast so that the missile hardbody can be detected in a sequence frame, without the need for individual sample correlations. Therefore, PC is quite fast and computationally efficient when applied to image correlation.

An overview of the video processing system is shown in the Figure **??**, in which the stabilization algorithm provides a preprocessed sequence to the PC algorithm. Section **??** describes the passive image stabilization algorithm developed to register missile video sequences. PC, as presented in Section **??**, can then be applied to the stabilized data. PC is defined as a first-order nonlinear recursion, which can efficiently obtain collective higher-order correlations from the data. Statistical properties of the technique are discussed. Experimental results are given for both synthetic and digitized video sequences in Section **??**. A summary of the research is provided in Section **??**.

II. Image Stabilization

In a general sense, image stabilization involves a registration of the frames within a video sequence, so that moving objects appear stationary after processing. Techniques of motion estimation may be applied to the problem, estimating the displacement of each pixel between adjacent frames [Musmann85]. Once displacement vectors have been calculated, the motion is compensated. By applying this procedure to all video frames, a stabilized sequence results containing ideally no movement. However, the accurate estimation of displacement vectors for all image pixels requires a great deal of computation. If pertinent features exist in the data, they may serve as reference points which can be tracked through the sequence. In addition, these *stable features* may be used to stabilize the sequence, by assuming all motion within the data occurs with respect to those features. A fully-parametric motion estimation algorithm results, in which motion can be completely described by a small number of parameters associated with a data model [Anandan93].

Missile image sequences acquired in the infrared spectrum contain a plume emanating from the missile, which appears in image data as an elliptical shape. A frame from such a sequence is shown in Figure ??. Thermal intensity is greatest near the nozzle of the missile, in a portion of the plume known as the vacuum core. A single frame from a missile sequence may contain multiple pixels belonging to the vacuum core. If the infrared data is viewed as a surface, the peak of the plume is found directly beneath the missile. A surface plot of the plume containing the vacuum core pixels is also shown in Figure ??. The vacuum core may be considered as a stable feature within the data, if the missile plume can be easily distinguished from the surrounding data.

Passive image stabilization tracks the vacuum core, using data near the plume peak as the reference point for all motion within the sequence. Thus, as the vacuum core undergoes translation from frame to frame, all pixels belonging to the plume and the missile hardbody translate with the same direction and displacement. As the missile rotates or changes scale in the sequence, all samples undergo the same transformation as well. In this research, only translation and rotation of the missile will be considered. By properly transforming the data, a sequence results with the missile in the same position in each frame.

A. Passive Image Stabilization

The flowchart in Figure ?? gives an overview of the passive image stabilization algorithm. The nonlinear subsystem serves essentially two purposes: a least squares surface fit is used to estimate the position of the vacuum core within the data, while morphological processing is used to estimate missile orientation. The linear subsystem is a Kalman filter, used to smooth the position of the stable feature in time. In this section, the translation and rotation parameters will be defined, by examining the spatial coordinate system assumed for a sequence frame. A data transformation matrix results, which may be used to translate and rotate all samples within a frame. The image stabilization algorithm is sequential in nature, so that a real-time implementation is possible.

A. Stabilization Parameters and Data Transformation

The assumed coordinate system for a single frame is shown in Figure ??, with the estimated line through the center of the missile plume given as $v = \alpha u + \beta$. The origin has been placed at the image center, and the position of the estimated vacuum core is designated by coordinates (t_u, t_v). The estimated slope of the line passing through the center of the plume is used to determine the missile rotation angle, $\theta_0 - \theta = \theta_0 - \tan^{-1}\alpha$. In this way, the missile can be rotated to some arbitrary angle θ_0 with respect to the coordinate axes. Unknown parameters include the data translation coordinates (t_u, t_v) and rotation angle $\theta_0 - \theta$.

Homogeneous coordinates have been used to determine the position of the translated and rotated image pixels [Foley90]. The advantage of this coordinate system is that translation, scaling, and rotation matrices can be multiplied to obtain a composite transformation matrix. The new pixel position P^* is given by the composite matrix equation

$$P^* = R(\theta_0 - \theta)\ T(-t_u, -t_v)\ P, \tag{1}$$

with $T(-t_u, -t_v)$ initially translating the vacuum core to the origin, and $R(\theta_0 - \theta)$ rotating the data about the vacuum core through an angle $\theta_0 - \theta$. Using the homogeneous coordinate system translation and rotation matrices defined by Foley *et al.* [Foley90], the new position P^* becomes

$$\begin{bmatrix} u^* \\ v^* \\ 1 \end{bmatrix} = \begin{bmatrix} (u - t_u)\cos(\theta_0 - \theta) + (-v + t_v)\sin(\theta_0 - \theta) \\ (v - t_v)\cos(\theta_0 - \theta) + (u - t_u)\sin(\theta_0 - \theta) \\ 1 \end{bmatrix}. \tag{2}$$

All pixels within a frame are translated and rotated using this composite transformation matrix. Interpolation may be necessary if the data undergoes rotation, since individual pixel positions may not coincide with the image array. Bilinear interpolation is used in practice, although higher-order methods such as cubic spline and Lagrange interpolation could be incorporated [Jain89]. If the missile rotates slowly throughout the sequence, then data rotation is not necessary for the successful enhancement of the missile hardbody signal by product correlation.

B. Vacuum Core Estimation by Least Squares Surface Fit

Passive stabilization is dependent on the vacuum core pixels of the missile plume. For infrared data, the peak pixel of the missile signature plume should be within the set of vacuum core pixels for a particular frame. Accurate detection of this peak pixel is necessary for the algorithm to perform correctly.

Missile video sequences may contain stars in the background or corrupted pixels due to malfunctioning image sensors. These artifacts appear as impulsive noise in a video frame. A 3×3 median filter is used to preprocess each frame to rid the data of such extraneous peaks. The peak pixel location can then be detected accurately.

Since the peak pixel is assumed to be a member of the set of vacuum core pixels, a least squares surface fit to data near the peak can be used to estimate the vacuum core location in the presence of noise or missing data. A square neighborhood of samples surrounding the peak pixel will be used to compute the surface fit, and the location of the resulting surface peak will be the vacuum core position estimate. Assume that the peak pixel value, $p(u, v)$, is centered within the neighborhood

$$\mathcal{N}(u, v) = \{p(k, l) \ : \ u - d \leq k \leq u + d, v - d \leq l \leq v + d\}. \tag{3}$$

In this expression, d represents the extent of a $(2d + 1) \times (2d + 1)$ pixel data window. An estimated surface value, $\hat{p}(k,l)$, is then given by the quadric polynomial

$$\hat{p}(k,l) = a_1k^2 + a_2l^2 + a_3kl + a_4k + a_5l + a_6, \tag{4}$$

where coefficients $\{a_i\}_{i=1}^{6}$ are determined from samples contained within $\mathcal{N}(u,v)$.

Coefficients can be estimated by minimizing the squared error between the observed pixel $p(k,l)$ and the estimated surface value $\hat{p}(k,l)$. Let $a \in \mathbb{R}^6$ represent the six polynomial coefficients as a vector. To calculate the least squares surface coefficients, differentiate the right hand side of the expression

$$\hat{a} = \arg\min_{a} \sum_{k=u-d}^{u+d} \sum_{l=v-d}^{v+d} \left[p(k,l) - a_1k^2 - a_2l^2 - a_3kl - a_4k - a_5l - a_6\right]^2 \tag{5}$$

with respect to each of the coefficients, a_i, for $i = 1, \ldots, 6$, setting these six equations equal to zero. A positive definite system of normal equations results, in which the surface coefficients can be calculated by an efficient Gaussian elimination algorithm without row or column interchanges [Kincaid91].

An example of this procedure is depicted in Figure **??**. A data window consisting of 15×15 samples centered at the peak pixel $p(u,v)$ was extracted from the infrared video frame in Figure **??**. The least squares surface was calculated using this data, shown in Figure **??**. The location of the surface peak, $\hat{p}(t_u, t_v)$, is taken as an initial estimate of the data translation parameters.

C. Orientation Estimation by Morphological Processing

Morphology deals with the study of object structure within images. Morphological processing refers to operations on an object within an image, in which a structuring element is used to reduce an object to a more revealing shape. The morphological operation of skeletonization [Jain89] is used to estimate the orientation of the missile. The skeleton is the elemental form of an object, a single pixel width representation which maintains the basic identity. The missile plume skeleton is a straight line, passing through the center of the plume. This skeleton can be used to estimate missile orientation.

Skeletonization makes use of the opening operation, which incorporates an erosion of the data. Initially, a sequence frame is binarized by thresholding the data to isolate the missile plume as an object. A 3×3 square grid is used as the morphological structuring element, translated through the binary sequence frame to erode the plume into a straight line. This skeleton passes directly through the center of the plume. The line equation, $v = \alpha u + \beta$, is then fit to the missile plume skeleton using a least squares technique [Haralick92]. The rotation parameter becomes a simple function of the line slope, $\theta_0 - \theta = \theta_0 - \tan^{-1} \alpha$.

D. Kalman Filtering

Use of a Kalman filter to estimate missile dynamics in a video sequence can provide a smoother estimate of the vacuum core location than that provided by the least squares surface fit alone. Kalman filtering will not be applied to the orientation parameter θ, since morphological skeletonization provides an accurate estimate of this parameter. Assume that the discrete dynamic model has the form

$$f_{k+1} = \Phi f_k + \eta_k, \tag{6}$$

where $f_k \in \mathbb{R}^q$ is the state vector containing q states for image frame k, Φ is a linear, time-invariant state transition matrix with dimension $q \times q$, and $\eta_k \in \mathbb{R}^q$ is zero mean Gaussian noise with covariance matrix Q_k, representing uncertainty in the state estimate.

To derive the form of the state transition matrix, the missile dynamics must be determined. Assume that the true (unknown) position of the missile vacuum core at time t is given by the vector $(u(t), v(t))$, with unknown translational velocity $(\dot{u}(t), \dot{v}(t))$. The state vector of unknown model parameters is defined as

$$f(t) = \begin{bmatrix} u(t) \\ \dot{u}(t) \\ v(t) \\ \dot{v}(t) \end{bmatrix}. \tag{7}$$

Assuming that the missile exhibits a constant velocity, the differential equation

$$\dot{f}(t) = \begin{bmatrix} \dot{u}(t) \\ 0 \\ \dot{v}(t) \\ 0 \end{bmatrix} \tag{8}$$

applies, with arbitrary initial conditions $u(t)$ and $v(t)$. Integrating this differential equation from t to $t+T$ generates the following difference equations [Broida91]:

$$\begin{aligned} u(t+T) &= u(t) + T\dot{u}(t) \\ \dot{u}(t+T) &= \dot{u}(t) \\ v(t+T) &= v(t) + T\dot{v}(t) \\ \dot{v}(t+T) &= \dot{v}(t) \end{aligned}$$

The index k will be used for the time index corresponding to the k^{th} frame within the sequence, so that the discrete dynamic model becomes

$$\begin{bmatrix} u_{k+1} \\ \dot{u}_{k+1} \\ v_{k+1} \\ \dot{v}_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_k \\ \dot{u}_k \\ v_k \\ \dot{v}_k \end{bmatrix} + \eta_k, \tag{9}$$

explicitly showing the state transition matrix Φ. For video rate data, the period between frames is $T = \frac{1}{30}$ seconds.

The observation model for the missile tracking system has the form

$$g_k = H f_k + n_k, \tag{10}$$

where $g_k \in \mathbb{R}^2$ is the observed position of the vacuum core, $H \in \mathbb{R}^{2\times 4}$ is a measurement matrix, and $n_k \in \mathbb{R}^2$ is zero mean Gaussian noise with covariance matrix R_k. Explicitly, this model is given as

$$g_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u_k \\ \dot{u}_k \\ v_k \\ \dot{v}_k \end{bmatrix} + n_k. \tag{11}$$

In this expression, the observed vacuum core position g_k is an initial estimate of the translation parameters t_u and t_v for the k^{th} frame, obtained by the least squares surface fit.

The Kalman filter dynamically updates the state f_k using all former state estimates to improve the solution. The derivation of the Kalman equations for discrete-time systems is well-documented [Grewal93], [Maybeck79]. In each recursion of the Kalman algorithm, both the state estimate f_{k+1} and the error covariance P_{k+1} are first predicted from previous iteration values, and then corrected by the Kalman gain matrix. Each frame within the image sequence corresponds to a state iterate, with the k^{th} state estimate providing the translation parameters $t_u = \hat{u}_k$ and $t_v = \hat{v}_k$ for the k^{th} frame. Convergence is determined by examining the state error covariance matrix, denoted as $P_k = E[(f_k - \hat{f}_k)(f_k - \hat{f}_k)']$.

Once the translation and rotation parameters are estimated for a frame, all pixels are transformed using (**??**). After stabilization of the missile hardbody, temporal correlations can be estimated through the use of product correlation.

III. Product Correlation

Assume that a set of stabilized data is available from the algorithm described in Section 2. Conventional second-order statistics (covariance) may be applied to the data to compute the temporal correlations. However, HOS may contain information not included within the second-order statistics. For instance, HOS are used for nonminimum phase system identification [Giannakis89], [Lii82], non-Gaussian modeling [Giannakis90], harmonic decoupling problems [Nikias87], and blind identification [Tong91]. To incorporate HOS, the higher-order moments or cumulants must be computed. Generally, the computational burden associated with HOS is much greater than that of second-order statistics. When we compute the temporal correlations in the stabilized data, we compute second-, third-, and fourth- or even higher-order correlations individually, with the summation of terms of each order representing the complete statistical information. However, this requires an enormous amount of computation.

Product correlation offers a computationally efficient means of obtaining sample higher-order correlations collectively. It is recursive in nature and yields collective sample higher-order correlations in one variable. This method can be viewed as a multilinear transformation of the input data. In this section, we will define PC, examining statistical properties and computational efficiency.

A. Definition

Let a sequence of images be denoted as $\{x_i(u,v)\}$, where u and v are row and column indices in the spatial domain, and i is the time index in the temporal domain. To simplify notation, the spatial domain indices will be dropped, so that x_i denotes $x_i(u,v)$.

Definition 1 (Product Correlation) *The correlated output sequence of images, $\{y_i\}$, is generated by the recursive relation*

$$y_i = y_{i-1}x_i + x_i + y_{i-1}. \tag{12}$$

This is a nonlinear time series model. However, it is in the special class of nonlinear time series models known as *bilinear time series models*, which are encountered in control systems and economic time series analysis. PC can be viewed as the first-order bilinear time series model. Stability analysis and parameter estimation of bilinear time series models were investigated by Granger [Granger78] and Rao [Rao84]. In this paper, we will address computational efficiency and statistical information contained within PC. A schematic block diagram of PC is shown in Figure **??**.

Without loss of generality, the initial condition y_0 is often set to zero. Straightforward algebraic manipulation of the definition in (**??**) reveals

$$\begin{aligned}
y_1 &= x_1 \\
y_2 &= x_1x_2 + x_1 + x_2 \\
y_3 &= x_1x_2x_3 + x_1x_2 + x_1x_3 + x_2x_3 + x_1 + x_2 + x_3 \\
y_4 &= x_1x_2x_3x_4 + x_1x_2x_3 + x_1x_2x_4 + x_1x_3x_4 \\
&\quad + x_2x_3x_4 + x_1x_2 + x_1x_3 + x_1x_4 + x_2x_3 + x_2x_4 + x_3x_4 \\
&\quad + x_1 + x_2 + x_3 + x_4
\end{aligned} \tag{13}$$

and so forth.

The correlated output of PC contains a sum of product terms, which can be considered as the sum of sample correlations. A closer examination of (??) leads to the following general expression. The correlated output at the N^{th} iteration, y_N, is the sum of collective r^{th} order correlations, m_{rx}, for $r = 1, \ldots, N$,

$$y_N = \sum_{r=1}^{N} m_{rx}, \tag{14}$$

where m_{rx} is defined as

$$m_{rx} = \underbrace{\sum_{j_1=1}^{N-r+1} \sum_{j_2=j_1+1}^{N-r+2} \cdots \sum_{j_r=j_{r-1}+1}^{N}}_{r} x_{j_1} x_{j_2} \ldots x_{j_r}. \tag{15}$$

For an intuitive understanding of (??), reexamine the first-order correlation

$$m_{1x} = \sum_{j_1=1}^{N} x_{j_1}. \tag{16}$$

If we introduce the scaling factor $\frac{1}{N}$, then (??) becomes the sample mean. The collective second and third-order correlations can also be obtained from (??) as

$$m_{2x} = \sum_{j_1=1}^{N-1} \sum_{j_2=j_1+1}^{N} x_{j_1} x_{j_2} \tag{17}$$

$$m_{3x} = \sum_{j_1=1}^{N-2} \sum_{j_2=j_1+1}^{N-1} \sum_{j_3=j_2+1}^{N} x_{j_1} x_{j_2} x_{j_3}. \tag{18}$$

Thus, as i becomes very large, y_i can be used as an approximation to the correlation sum of all orders. Evaluations of y_i reveal critical information regarding higher-order correlations of x_i. It has been shown that PC provides the collective higher-order correlations recursively in one variable. Therefore, we can obtain the HOS information from the correlated output of PC instead of computing individual higher-order correlations. For the contrast enhancement of missile video sequences, we do not need to compute individual correlations. Only collective statistical information is required. From this point of view, PC is a powerful method for computing temporal correlations.

B. Statistical Information

From (**??**) and (**??**), it can be seen that PC contains sample correlations of various orders. To verify the statistical content, we will examine a scaled version of PC.

Definition 2 (Scaled product correlation) *By introducing the time index i as a scaling factor, PC can be modified as*

$$\begin{aligned}\tilde{y}_i &= \frac{1}{i} y_i \\ &= \frac{1}{i}\{y_{i-1}x_i + x_i + y_{i-1}\}.\end{aligned} \tag{19}$$

The block diagram of this modified product correlation with scaling factor is shown in Figure **??**. We are going to consider the scaled correlated output, $\tilde{y}_i$ to obtain a relationship with higher-order sample correlations.

Now, we will examine biased correlation estimates. Suppose data set $\{x_1, x_2, \ldots, x_N\}$ is available. The second-order correlation at lag m, and the third-order correlation at lags $\{m, n\}$ are

$$\begin{aligned}R_2(m) &= E\{x_i x_{i+m}\} \\ R_3(m,n) &= E\{x_i x_{i+m} x_{i+n}\}.\end{aligned} \tag{20}$$

The biased estimates of the second- and third-order correlations with N data samples are

$$\hat{R}_2^{(N)}(m) = \frac{1}{N}\sum_{i=t_1}^{t_2} x_i x_{i+m}$$

$$\hat{R}_3^{(N)}(m,n) = \frac{1}{N}\sum_{i=s_1}^{s_2} x_i x_{i+m} x_{i+n}, \tag{21}$$

where

$$\begin{aligned} t_1 &= max(1, 1-m) \\ t_2 &= min(N, N-m) \\ s_1 &= max(1, 1-m, 1-n) \\ s_2 &= min(N, N-m, N-n). \end{aligned}$$

The following example shows how the modified PC expression in (**??**) provides the sum of biased sample correlation estimates. For simplicity, let $\{x_1, x_2, x_3\}$ be a zero mean data set. The biased estimates of the first (mean), second, and third-order correlations are

$$\begin{aligned} \mu_3 &= \frac{1}{3}\{x_1 + x_2 + x_3\} \\ \hat{R}_2^{(3)}(1) &= \frac{1}{3}\{x_1 x_2 + x_2 x_3\} \\ \hat{R}_2^{(3)}(2) &= \frac{1}{3}\{x_1 x_3\} \\ \hat{R}_3^{(3)}(1,2) &= \frac{1}{3}\{x_1 x_2 x_3\}. \end{aligned}$$

The correlated output of scaled PC when $i = 3$ is

$$\tilde{y}_3 = \frac{1}{3}\{x_1 x_2 x_3 + x_1 x_2 + x_2 x_3 + x_1 x_3 + x_1 + x_2 + x_3\}, \tag{22}$$

so that $\tilde{y}_3$ can be rewritten as

$$\tilde{y}_3 = \hat{R}_3^{(3)}(1,2) + \hat{R}_2^{(3)}(1) + \hat{R}_2^{(3)}(2) + \mu_3. \tag{23}$$

Therefore, PC provides the sum of the biased correlation estimates. By using this notation, we can rewrite the general expression of the correlated output of scaled PC, $\tilde{y}_N$, in terms of the biased correlation estimates as

$$\tilde{y}_N = \sum_{r=1}^{N} \tilde{m}_{rx}, \tag{24}$$

where

$$\tilde{m}_{rx} = \underbrace{\sum_{l_1=1}^{N-r+1} \sum_{l_2=l_1+1}^{N-r+2} \cdots \sum_{l_{r-1}=l_{r-2}+1}^{N-1}}_{r-1} (N - l_{r-1}) \hat{R}_r^{(N)}(l_1, l_2, \ldots, l_{r-1}). \tag{25}$$

This representation provides a clarification of the type of statistical information inherent within PC.

C. Presence of Additive Noise

Let the observed data x_i be the sum of the signal s_i and the background noise w_i,

$$x_i = s_i + w_i. \tag{26}$$

The signal and noise are assumed to be statistically independent, and the noise is independent and identically distributed (i.i.d.) with zero mean. For example, we have

$$\begin{aligned} E\{f(s_i)g(w_j)\} &= E\{f(s_i)\}E\{g(w_j)\} \\ E\{f(w_i)g(w_j)\} &= E\{f(w_i)\}E\{g(w_j)\} \quad \text{for any f and g, } i \neq j \end{aligned} \tag{27}$$

Assuming zero initial conditions, the output y_i can be expressed as follows:

$$\begin{aligned} y_1 &= s_1 + w_1 \\ y_2 &= s_1 s_2 + s_1 + s_2 + w_1 w_2 + w_1 + w_2 + \text{crossterms} \end{aligned}$$

$$
\begin{aligned}
y_3 &= s_1s_2s_3 + s_1s_2 + s_1s_3 + s_2s_3 + s_1 + s_2 + s_3 \\
&\quad + w_1w_2w_3 + w_1w_2 + w_1w_3 + w_2w_3 + w_1 + w_2 + w_3 + \text{crossterms}
\end{aligned}
$$

Crossterms are products of the signal s_i and noise w_i. If y_n^x denotes the correlated sequence after passing x through PC when i is equal to n, then

$$
y_n^x = y_n^s + y_n^w + \text{crossterms}. \tag{28}
$$

Taking the expectation of y_n^x results in the expression

$$
\begin{aligned}
E\{y_n^x\} &= E\{y_n^s + y_n^w + \text{crossterms}\} \\
&= E\{y_n^s\} + E\{y_n^w\}.
\end{aligned} \tag{29}
$$

Expectation of the crossterms tends to zero since the signal and noise are statistically independent and w_i is i.i.d. with zero mean. For example, let us denote the crossterms in y_2^x as y_2^{sw}, then

$$
y_2^{sw} = s_1s_2w_3 + s_1s_3w_2 + s_1w_2w_3 + s_2s_3w_1 + w_1w_3s_2 + w_1w_2s_3
$$

The expectation of y_2^{sw} is

$$
\begin{aligned}
E\{y_2^{sw}\} &= E\{s_1s_2w_3 + s_1s_3w_2 + s_1w_2w_3 + s_2s_3w_1 + w_1w_3s_2 + w_1w_2s_3\} \\
&= E\{s_1s_2\}E\{w_3\} + E\{s_1s_3\}E\{w_2\} + E\{s_1\}E\{w_2\}E\{w_3\} \\
&\quad + E\{s_2s_3\}E\{w_1\} + E\{w_1\}E\{w_3\}E\{s_2\} + E\{w_1\}E\{w_2\}E\{s_3\}
\end{aligned}
$$

We can easily see the expectation of these crossterms are zero according to the independence assumption between signal and i.i.d. noise with zero mean. Also, since w_i is i.i.d. with zero mean,(**??**) simplifies to

$$E\{y_n^x\} = E\{y_n^s\}. \tag{30}$$

From this expression, it can be inferred that the ensemble average of filtered data with PC is due to the signal exclusively. If the background mean is not equal to zero, the bias term in (**??**) may affect PC. Therefore, the mean of the background must be subtracted to remove the bias. For a nonzero mean signal, (**??**) can be modified as

$$y_i = y_{i-1}(x_i - \mu_i) + (x_i - \mu_i) + y_{i-1}, \tag{31}$$

where μ_i is the sample mean of w_i in the temporal domain. However, it is impossible to compute the sample temporal mean of w_i in the plume area. Thus we replace the temporal mean by spatial mean. In other words μ_i is the sample spatial mean of background noise w_i. If the background is spatially nonstationary, spatial mean is not a good approximation of temporal mean. Therefore background noise is not zero mean anymore, which affects the object resolution.

D. Perfect Stabilization

In a practical sense, any video stabilization algorithm will not produce a perfectly stabilized sequence. However, it is worthwhile to investigate the filtered output of s_i through PC under an assumption of perfect stabilization.

Assuming that every frame of a sequential image data set is perfectly stabilized, the observed data x_i can be described as

$$\begin{aligned} x_1 &= s + w_1 \\ x_2 &= s + w_2 \end{aligned}$$

$$
\begin{aligned}
&\vdots \quad && \vdots \\
x_N &= && s + w_N.
\end{aligned}
$$

Intuitively, this indicates that s is not changed, which means that it is simply a deterministic signal. The fact that the signal is not changed in every frame brings the following relations:

$$
\begin{aligned}
y_1^s &= s \\
y_2^s &= s^2 + 2s \\
y_3^s &= s^3 + 3s^2 + 3s \\
y_4^s &= s^4 + 4s^3 + 6s^2 + 4s
\end{aligned} \tag{32}
$$

From (**??**), we can deduce the general representation for y_N^s,

$$
y_N^s = \sum_{k=1}^{N} \begin{pmatrix} N \\ k \end{pmatrix} s^k \tag{33}
$$

where

$$
\begin{pmatrix} N \\ k \end{pmatrix} = \frac{N!}{k!(N-k)!}. \tag{34}
$$

We have thus derived a simpler form for the PC output to the signal s under the assumption of perfect stabilization. The general form can be found in (**??**) and (**??**).

E. Signal-to-Noise Ratio

The goal of PC is to enhance the contrast between the object and the background in order to track the missile hardbody. SNR serves as a good measure of contrast enhancement. By computing the ratio of the correlated output to the signal, s, and the correlated output to the noise, w, the

SNR becomes

$$\frac{y_N^s}{y_N^w} = \frac{m_{rs}^N}{m_{rw}^N}, \tag{35}$$

where

$$\begin{aligned}
m_{rs}^N &= \underbrace{\sum_{j_1=1}^{N-r+1} \sum_{j_2=j_1+1}^{N-r+2} \cdots \sum_{j_r=j_{r-1}+1}^{N}}_{r} s_{j_1} s_{j_2} \ldots s_{j_r} \\
m_{rw}^N &= \underbrace{\sum_{j_1=1}^{N-r+1} \sum_{j_2=j_1+1}^{N-r+2} \cdots \sum_{j_r=j_{r-1}+1}^{N}}_{r} w_{j_1} w_{j_2} \ldots w_{j_r}.
\end{aligned}$$

Taking the expected value of both the numerator and denominator in (**??**) results in

$$\frac{E\{y_N^s\}}{E\{y_N^w\}} = \frac{\sum_{j_1=1}^{N-r+1} \sum_{j_2=j_1+1}^{N-r+2} \cdots \sum_{j_r=j_{r-1}+1}^{N} E\{s_{j_1} s_{j_2} \ldots s_{j_r}\}}{\sum_{j_1=1}^{N-r+1} \sum_{j_2=j_1+1}^{N-r+2} \cdots \sum_{j_r=j_{r-1}+1}^{N} E\{w_{j_1} w_{j_2} \ldots w_{j_r}\}}. \tag{36}$$

Theoretically, SNR as defined in (**??**) tends to infinity since noise was assumed to be independent with zero mean, and signal correlations in the numerator increase in magnitude. In other words, PC produces the maximum theoretical SNR iteratively. In practice, however, PC is an approximation to the ensemble statistics, and the denominator in (**??**) is not exactly zero because we can not obtain the exact mean of the background. Since the subtracted-spatial-mean noise can not be zero mean in the presence of spatial nonstationarity, signal $\times$ noise and noise $\times$ noise terms would grow as signal power increaes, but would be much dimmer than the signal $\times$ signal terms. This explains why PC enhances the contrast between the signal and noise. Additionally, PC can suppress i.i.d. zero mean noise, since noise correlations are zero.

IV. Experimental Results

To implement PC for an image sequence, PC is applied to each pixel separately in every frame. Therefore, a pixel located in a particular row and column throughout all image frames can be considered as a random process. For efficient computation, PC could easily be parallelized. For instance, consider a sequence in which each frame contains $N \times N$ pixels. Up to $N \times N$ processors could perform computations in parallel on such a data set, providing linear speed-up.

The PC implementation flowchart is shown in Figure **??**. A stabilized image sequence must be available as input to the PC algorithm, which consists of the following steps:

1. Import a single frame of the stabilized sequence, x_i.
2. Compute the spatial sample mean μ_i of the noisy background to remove the background bias.
3. Compute the correlated output using (**??**).
4. Saturate the intensity of each pixel at a predefined level.
5. Retain the value of the correlated output for the next iteration.
6. Return to Step 1 until all frames have been processed.

Step 4 within the algorithm requires further explanation. The intent is to increase the intensity of the missile hardbody signal. The intensity of the hardbody signal will be enhanced compared to noisy background pixels by computing the temporal correlations, but the plume intensity will still dominate an image frame. By saturating the plume intensity at a specified level, we can enhance the missile hardbody intensity as well.

A. Synthetic Video Sequence

A synthetic image sequence which contains a known missile signal, a missile plume with constant intensity, and a uniform background was generated to examine how well a missile signal is enhanced in the presence of additive white Gaussian noise. Noise is uncorrelated in the temporal domain, so that the noise processes are uncorrelated between frames at each pixel. SNR is computed by the expression

$$\text{SNR} = 20 \log \left(\frac{C}{\sigma} \right), \tag{37}$$

where C is missile intensity, and σ is the standard deviation of the Gaussian noise. Figure **??** shows the input image and the correlated output for the 5^{th}, 6^{th}, and 7^{th} iterations of PC. The SNR in the input image was 9.5 dB. The minimum saturation bound was set at -10.0 and the maximum bound was set at 1.0×10^{11}.

B. Digitized Video Sequences

Passive stabilization was used to stabilize two digitized missile video sequences provided by Rome Laboratories, Rome, NY. A frame from the first sequence is shown in Figure **??**, which was acquired in the infrared region of the spectrum. A portion of a frame from the second sequence is shown in Figure **??**, which was acquired in the visible portion of the spectrum. Both digitized sequences were stabilized successfully using the algorithm presented in Section 2.

Experimental PC will concentrate on the video sequence acquired in the visible spectrum, shown in Figure **??**. This data set contains a visible hardbody, so that PC implementation results can be verified. In the simulations, minimum and maximum saturation bounds were set at -1.0×10^3 and 1.0×10^{20}, respectively.

With the visible data set, three type of simulations were performed. First, second-order sample correlations were used to enhance the hardbody, with results shown in Figure **??**. Next, PC was applied to the data set, with the correlated output shown in Figures **??** and **??**. Finally, ensemble averaging of data blocks was applied to obtain a better filtered output than regular PC. To obtain the ensemble average, the data was segmented into several data records, correlated output was computed for each data record, and all records were averaged. Let the number of sequences available be M, and divide this into K records of N samples each. Denoting $y_i^{(K)}$ as the correlated output from the K^{th} data record, the averaged correlated output y_i is given as

$$y_i = \frac{1}{K}\sum_{l=1}^{K} y_i^{(l)}. \tag{38}$$

Diagrams describing this method are shown in Figures ?? and ??. Averaging the correlated output from each data block provides a better estimate of correlation than the correlation estimate obtained by original PC. Figures ?? and ?? show the ensemble average of correlated output over 5 data records and 10 data records, respectively. Note that ensemble averaging provides better results with some smoothing. However, it is not feasible for a real-time implementation, since a partitioning of the sequence is required.

As an alternative to the ensemble averaging technique, an *interleaving method* may be applied to the sequence, resulting in a viewable sequence containing a missile hardbody of relatively constant intensity. The basic concept of the interleaving method is shown in Figure ??. PC is performed on a block of frames, with frames containing an enhanced missile signal viewed as a subsequence of the complete interleaved PC sequence.

The SNR was computed to demonstrate how much improvement PC provides. SNR is defined in this case as

$$\text{SNR} = \frac{\frac{1}{N_s}\sum_{i,j} s^2(i,j)}{\frac{1}{N_w}\sum_{i,j} w^2(i,j)}, \tag{39}$$

where $s(i,j)$ is a pixel representing the hardbody, $w(i,j)$ is a pixel representing the background, N_s is the number of pixels belonging to the hardbody, and N_w is the number of pixels belonging to the background. Before the hardbody is enhanced, the position of the hardbody is not known. After a predetermined number of iterations, PC sufficiently enhances the missile signal. Results were examined at the 11^{th} iteration to locate the hardbody signal. The missile signal was isolated within the PC output at this iteration, and individual samples were either assigned to the set of hardbody pixels if the PC output exceeded a predetermined threshold, or to the set of background pixels if the PC output fell below the threshold. The signal power is calculated for all PC iterations given the set of hardbody pixels, and similarly the noise power is calculated given the set of background pixels. Table **??** shows the resulting SNR for a number of PC iterations with two different threshold. The SNR of input stabilized image was around 1 dB. Thus we can see we are able to enhance the SNR by 20 dB in this case.

V. Summary

PC has been successfully applied to enhancing the contrast of a missile signal in video sequences exhibiting low SNR. Prior to performing PC on the data, a passive stabilization algorithm pre-processes the video sequence so that the missile signal is in the same position within each frame. Passive stabilization is dependent upon a stable feature within the data, with the missile vacuum core selected in the algorithm. Experimentally, the algorithm has successfully stabilized both infrared and visible video sequences. A theoretical analysis and a practical PC implementation have attempted to verify its properties and potential. Iteratively, it provides the sum of collective higher-order correlations in a single variable. A great deal of statistical information is obtained without the computational effort of calculating individual sample correlations. For both synthetic and digitized video sequences, PC was successful in locating the missile hardbody in low SNR environments. A key point of both the passive stabilization and PC algorithms is that all steps are of a sequential nature. Thus, the algorithms could be implemented in real-time, by a proper mapping into computer hardware.