Infrared Image Enhancement using $H_\infty$ Bounds for Surveillance Applications

Uvais Qidwai, Member, IEEE

Abstract—In this paper, two algorithms have been presented to enhance the infrared (IR) images. Using the autoregressive moving average model structure and $H_\infty$ optimal bounds, the image pixels are mapped from the IR pixel space into normal optical image space, thus enhancing the IR image for improved visual quality. Although $H_\infty$-based system identification algorithms are very common now, they are not quite suitable for real-time applications owing to their complexity. However, many variants of such algorithms are possible that can overcome this constraint. Two such algorithms have been developed and implemented in this paper. Theoretical and algorithmic results show remarkable enhancement in the acquired images. This will help in enhancing the visual quality of IR images for surveillance applications.

Index Terms—$H_\infty$ identification, image enhancement, image modeling, infrared (IR) image processing.

I. INTRODUCTION

MAGES received through various infrared (IR) devices in many applications are distorted due to the atmospheric aberration mainly because of atmospheric variations and aerosol turbulence [1], [2]. In this paper, new algorithmic strategies have been presented to enhance the visual quality of IR images. The idea is to model the IR image pixels as an input output system with IR image as the input and a “similar” optical image as the output. The image modeling is carried out using the usual system identification strategies. The system identification problem is to estimate a model of a system based on observed input-output data. Several ways to describe a system and to estimate such descriptions exist and are being used in various applications. The identification process amounts to repeatedly selecting a model structure, computing the best model in the structure, and evaluating this model’s properties to see if they are satisfactory. The cycle can be itemized as follows [3], [4].

1) Design the experiment to collect input-output data from the system to be identified.
2) Select and define a model structure (a set of candidate system descriptions) within which a model is to be found. This would mean the order and number of unknown coefficients be identified and tuned to fit the data.
3) Compute the best model in the model structure according to the input-output data and a given criterion of fit. In other words, fine tuning the coefficient values to get the most optimal values under a given optimality criterion.
4) If the model is good enough, then stop; otherwise, go back to Step 3 to try another model set. Possibly also try other estimation methods.

A. Basic Recursive Identification Algorithm

A typical recursive identification algorithm is

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)[y(t) - \hat{y}(t)].$$  \hfill (1)

Here, $\hat{\theta}(t)$ is the parameter estimate at time $t$, and $y(t)$ is the observed output at time $t$. Moreover, $\hat{y}(t)$ is a prediction of the value based on observations up to time $t-1$ and also based on the current model (and possibly also earlier ones) at time $t-1$. The gain $K(t)$ determines in what way the current prediction error affects the update of the parameter estimate. It is typically chosen as

$$K(t) = Q(t)\psi(t)$$ \hfill (2)

where $\psi(t)$ is (an approximation of) the gradient with respect to $\theta$ of $\hat{y}(t|\theta)$. Note that autoregressive model structures that correspond to linear regressions can be written as

$$y(t) = \psi(t)\theta_0(t) + \epsilon(t)$$ \hfill (3)

where the regression vector $\psi(t)$ contains old values of observed inputs and outputs, $\theta_0(t)$ represents the true description of the system, and $\epsilon(t)$ is the noise source.

A popular optimal choice of $Q(t)$ can then be computed from Kalman filter as follows:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)[y(t) - \hat{y}(t)]$$ \hfill (4)

$$\hat{y}(t) = \psi(t)\hat{\theta}(t-1)$$ \hfill (5)

$$K(t) = Q(t)\psi(t)$$ \hfill (6)

$$Q(t) = \frac{P(t-1)}{R_2 + \psi(t)^TP(t-1)\psi(t)}$$ \hfill (7)

$$P(t) = P(t-1) + R_1 - \frac{P(t-1)\psi(t)\psi(t)^TP(t-1)}{R_2 + \psi(t)^TP(t-1)\psi(t)}$$ \hfill (8)

where $R_2 = E[\epsilon(t)^2]$ (the variance of noise), and $R_1$ is an offset matrix which depends upon the technique to be used.

A second approach can be formulated by discounting old measurements exponentially, so that an observation $\tau$ samples old carries a weight that is $\lambda^\tau$ of the weight of the most recent
observation. This means that the minimizing function at time \( t \) is
\[
\sum_{k=1}^{t} \lambda^{t-k} e^2(k)
\]
where \( \lambda \) is a positive number slightly smaller than 1 (typically 0.97-0.955) and is called the forgetting factor. This criterion can be minimized exactly in the linear regression case giving the following:
\[
Q(t) = P(t) = \frac{P(t-1)}{\lambda + \psi(t)^T P(t-1) \psi(t)}
\]
\[
P(t) = \frac{1}{\lambda} \left( P(t-1) - \frac{P(t-1) \psi(t) \psi(t)^T P(t-1)}{\lambda + \psi(t)^T P(t-1) \psi(t)} \right).
\]

II. TRUE \( H_{\infty} \) VERSUS DIRECT \( H_{\infty} \)

In general least-square estimation problems, one minimizes the integral of the power spectral density estimation error. This minimization of the average error power or error variance might result in a relatively large error power in some frequency range. In many practical situations, however, there is significant uncertainty in the power spectral density of the noise. An appropriate criterion, as argued by Zames in [5], is the \( H_{\infty} \) form of the estimation error spectrum. This \( H_{\infty} \) problem is based on the minimization of the magnitude of the estimation error spectrum rather than the average power and leads to better filters compared to other solutions. The \( H_{\infty} \) optimization scheme leads to improved performance of the deconvolution process over existing least squares or \( H_2 \)-based solutions, thus leading to enhanced defect impulse response, thereby improving the defect identification.

Given a linear time-invariant (LTI) system with input \( u(t) \), output \( y(t) \), and a transfer function \( H(z) \), the \( H_{\infty} \) form is expressed in time domain as
\[
\|H\|_{\infty}^2 = \max_{\|u\| \leq 1} \left\{ \frac{\|y\|^2}{\|u\|^2} \right\} = \max_{\|u\| \leq 1} \left\{ \sum_{k=0}^{\infty} \frac{y^2(k)}{\sum_{k=0}^{\infty} u^2(k)} \right\}.
\]
Therefore, the \( H_{\infty} \) norm is interpreted as the maximum energy gain of the system for all finite energy signals. In the frequency domain, the \( H_{\infty} \) norm is expressed as
\[
\|H\|_{\infty} = \max_{\omega} |H(e^{j\omega})|
\]
where \( \omega \) is the largest peak of the frequency response magnitude.

Once the input-output model is formulated in the form of a transfer function, it can be related to an autoregressive moving average (ARMA) process as follows:
\[
D(z^{-1})y(k) = N(z^{-1})u(k) + E(z^{-1})
\]
where \( y(k) \) is the output of the process, \( u(k) \) is the input to the process, \( e(k) \) is the driving noise and
\[
D(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_{na} z^{-na}
\]
\[
N(z^{-1}) = b_0 + b_1 z^{-1} + \cdots + b_{nb} z^{-nb}
\]
where \( na \) and \( nb \) are the degrees of the polynomials \( D(z^{-1}) \) and \( N(z^{-1}) \), which are the denominator and numerator of the ARMA model, respectively. Model (14)-(16) is now put in a more suitable form for application of an identification algorithm
\[
y_k = \psi_k^T \theta + \epsilon_k
\]
where \( y_k \) denotes the samples of the output
\[
\psi_k = [-y(k+1) - y(k+2) \cdots -y(k-na) \ u(k) \ u(k-1)u(k-nb)]
\]
and
\[
\theta = [a_1 \ a_2 \ \cdots \ \ a_{na} \ b_0 \ b_1 \ \cdots \ b_{nb}].
\]
A linear parameter estimator \( T \) must be found to operate on such that the unbiased estimate at \( i = k \) is denoted by
\[
\hat{\theta}_k = T(y_i, u_i, i \leq k).
\]
The estimator \( T \) should be chosen for the worst possible \( e_k \) (representing noise and modeling errors), that produces a best estimate of the original image in the following sense. Denote
\[
e_{\gamma} = \gamma^{-1}(\theta - \hat{\theta}),
\]
Hence, the following differential-game problem can be minimized for a value of \( \gamma \) that is as close as possible to a minimum possible value \( \gamma_0 \)
\[
\sup_{e \in L^2[0 \cdots N]} \frac{\|\theta - \theta\|^2}{\|e_k\|^2 + \|\theta\|^2 R_0} < \gamma^2.
\]
Here, \( R_0 \) is the covariance matrix for the input-output data samples and the norm itself represents the realizable distance values within the variance of the data. Several algorithms have been reported in literature based on this reasoning, for example [6], [7]. As opposed to the classical \( H_{\infty} \) theory, there are alternatives that have been reported with similar bounds and optimal performance. These alternatives are especially attractive for this work since real-time processing is an essential part of the implementation. Expanding the usual time-based model of (3) to a sample-based 2-D domain, we can interpret the same quantities as imaging system. This is shown in Fig. 1.

Hence, the observed image \( y(m, n) \) for the actual input image \( u(m, n) \) is given by
\[
y(m, n) = u(m, n) \otimes h(m, n) + e(m, n)
\]
III. $H_{\infty}$ ALGORITHMIC FORMULATION FOR IMAGE ENHANCEMENT

The main objective in this paper is to develop an algorithmic structure which can map the convolution-based relationship between the true image and the distortion function, which, in this case, would be the hypothetical mapping function that maps the optical image to IR domain. In order to recover this original optical image, first, an inverse approach to image enhancement is formalized and then an appropriate algorithmic implementation is discussed in the following section. Quite opposite to the model in Fig. 1, the basic block diagram for image restoration given in Fig. 2 is an inverse look at the situation.

**Hypothesis:**

If the distortion function is taken as input to the true image, then output would result due to convolution of this function with the true image. Hence, by knowing the inverse parameters, essentially $B/A$ instead of $A/B$ would perform the inverse blurring (or de-blurring) operation if applied to the distorted image.

Since, the convolution is commutative in nature, the block diagram is modified for blur as input with original image as unknown block. The restoration system is represented by the following general linear dynamic model

$$y = \frac{B(z_1, z_2)}{A(z_1, z_2)} u + \frac{1}{A(z_1, z_2)} e$$  \hspace{1cm} (24)

where $y$ is the blurred image, $u$ is the blur function, $e$ is the additive noise, and $B/A$ corresponds to the original image transfer function. Similar models have been used in [8] and [9]. As such, the apparent blurring function, $h(m, n)$ is merely considered as an impulse function while all of its dynamics are mapped into the transfer function, $B/A$.

Equation (24) can also be written as

$$A(z_1, z_2)y = B(z_1, z_2)u + e.$$  \hspace{1cm} (25)

When expanded in 2-D series, this can be written as

$$y(n_1, n_2) = -\sum_{(k_1, k_2) \in R_a} \sum_{(k_1, k_2) \in R_b} a(k_1, k_2) y(n_1 - k_1, n_2 - k_2)$$

$$+ \sum_{(k_1, k_2) \in R_b} a(k_1, k_2) x(n_1 - k_1, n_2 - k_2) + e(n_1, n_2)$$

or, more compactly

$$y = \Phi \theta + e$$  \hspace{1cm} (27)

where $\Phi$ contains the input output measured samples, $\theta$ contains the parameters $a’s$ and $b’s$ to be estimated, and $R_a$ and $R_b$ are the real number spaces for the parameters. Hence, the image deconvolution scheme now reduces to finding out the actual distortion function in the form of a Transfer function and reconstruct the image using the estimated functions. This can be done through a variety of techniques and algorithms, however, the main emphasis in this work is to use the $H_{\infty}$ optimal identification for estimating the distortion models. Thus, the estimated models are supposed to be more robust and closer to the apparent real distortion function in comparison with the commonly used least-square estimator. However, the classical $H_{\infty}$ approach would be too complex and time consuming to be implemented for this purpose. An interesting finding presented in this work is the fact that there are indirect methods to obtain the optimal (or at least near optimal) performance of $H_{\infty}$ estimator by using less complicated methods under certain constraints where these simpler techniques converge to $H_{\infty}$ estimators. Putting it in simpler terms, this paper attempts to combine the simpler estimators under certain structural constraints to produce the performance of an $H_{\infty}$ estimator in order to produce better estimates of the distortion functions and consequently, by using inverse filtering, better images which are more visually suitable for surveillance applications. By avoiding the classical approach to $H_{\infty}$ estimation, a fast implementation for the same can be formulated and implemented in FPGA type hardware settings for real-time image enhancement. Based on the discussion in the Section 1-A, two approaches have been formulated in this paper that are categorized as two Algorithms and are outlined in the next section.

IV. PROPOSED ALGORITHMS

In this section, the developed algorithms are presented. The general structure for these algorithms is shown in Fig. 3.

For any particular camera settings, the environment is viewed in the beginning as a reference image. This is done in two ways.

1) Using the IR camera to capture the typical input image $(u(t))$. It is being called the input image here since after the initialization step, the real-time operation will only receive these images to enhance them.

2) Using a regular CCD type camera of approximately the same focal viewing properties as that of the IR camera collect the same images as in 1 simultaneously. This will constitute the target output $(y(t))$ for the imaging system. Note that this step will not work in dark since CCDs are light dependent.
This image will be used only in the initialization phase and will not be used in the real-time operation of the device.

Using the above-mentioned imaging pairs, the developed algorithms try to map the IR image structure into the photorealistic pixel space from the CCD camera, thus enhancing the IR image for visual applications.

A. Algorithm 1

This algorithm is based on the normalized LMS algorithm. The LMS algorithm is an approximate recursive estimator for solving the adaptive system identification problem; given \( N \times 1 \) input samples \( \{ u \} \) and corresponding numbers of samples of desired output \( \{ y \} \), the LMS estimator will try to find out a coefficient vector \( \theta \) such that the squared error sum \( \sum_{i=1}^{N} \| y_i - u_i \theta \|^2 \) is minimized. The LMS solution updates the coefficients recursively along the direction of the instantaneous gradient of the squared error. It has been shown in [10] that LMS under certain conditions obeys the \( H_\infty \) optimality criterion, and, hence, it can be used instead of the actual \( H_\infty \) algorithm. These conditions are listed as follows.

1) The input vectors should be exciting, i.e., \( \lim_{N \to \infty} \sum u_i \theta_i = \infty \) In other words, the degree of excitation in the input data samples should be sufficiently rich. This assumption is in general not true for IR images unless the noisy conditions are prevalent. However, this condition can be met by adding Gaussian random noise of very low amplitude to the input vectors.

2) Optimal gain in \( H_\infty \) terms for a strictly causal system is unity, i.e., \( \gamma_p = 1 \) for \( \| T \|_\infty < \gamma_p \), where \( T \) represents the transfer operator to be identified in order to map the input vector space into filtered space.

3) Central optimal \( H_\infty \) optimal \( a \) posteriori filter is given by

\[
\hat{\theta}_{j+1} = \hat{\theta}_j + \frac{\mu \Phi_j}{1 + \mu \Phi_j} (y_j - u_j \hat{\theta}_j). \tag{28}
\]

4) Central optimal \( H_\infty \) optimal \( a \) priori filter is given by

\[
\hat{\theta}_j = \hat{\theta}_{j-1} + \mu \Phi_j (y_j - u_j \hat{\theta}_{j-1}). \tag{29}
\]

Essentially, (28) is used in this algorithm under following architecture with \( \mu \), the learning rate, fixed at 0.9. 

1) Initialization:

1. Input the two simultaneous images, \( y(t) \) and \( u(t) \).
2. \( y(t) \) is stretched into a long vector by stacking its columns one after the other.
3. \( u(t) \) is stretched into 4 or 8 stacked columns representing the 4 or 8 Neighbors of the corresponding pixel in \( y(t) \). This arrangement artificially brings the concept of time samples from the spatial samples, as shown in Fig. 4. The 4 or 8-neighbor structures are very widely used in image processing algorithms and it is quite a straightforward procedure in hardware to expand an image into its neighboring pixel vectors. The structures are, therefore, used in order to make the hardware implementation transparent and in near-real time. It does limit the model order to 4 or 8. However, the order of 4 is used in this work as an appropriate compromise between speed and accuracy.

4) Using the stretched \( y(t) \) as output regressive process and stretched \( u(t) \) as input regressive process, corresponding number of coefficients are then provided to fit the estimates into an ARMA model. This model is given by

\[
y(t) + a_1 y(t-1) + \ldots + a_{na} y(t-na) \\
= b_{11} u_1 (t-1) + b_{12} u_1 (t-2) \\
+ \ldots + b_{1nb1} u_1 (t-nb1) \\
+ b_{21} u_2 (t-1) + b_{22} u_2 (t-2) + \ldots \\
+ b_{2nb2} u_2 (t-nb2) + \ldots \\
+ b_{Na} u_N (t-1) + b_{Na2} u_N (t-2) + \ldots \\
+ b_{NaN} u_N (t-nbN). \tag{30}
\]

5) Using the LMS algorithm, calculate the appropriate coefficients.

In this paper, the model used has \( N = 7, nb1 = nb2 = \ldots = nbN = 1 \), and \( a_1 = a_2 = \ldots = a_{na} = 1 \). Therefore, 11 coefficients overall are calculated as follows:

\[
\theta = [a_1, a_2, \ldots, a_7, b_{11}, b_{21}, b_{31}, b_{41}]^T. \tag{31}
\]

Various combinations of orders were tried and based on the mean squared error (MSE) between the estimated image using
the model parameters and the actual signal the structure in (31) was selected.

2) Real-Time Application (Reconstruction):
1) Once the coefficients are identified, they are used to enhance the new images within the same settings by convolving them, in the form of a mask, with the input image.
2) The mask formation is shown in Fig. 5. Essentially, it is the reverse of the stretching operation shown in Fig. 4.
3) The actual measured image is convolved with coefficients $b_i$’s. The resulting image is an improved (weighted and averaged) image ($y_1$ in the results).
4) An invert of the input image is obtained by subtracting the pixels of the input image from 255, thus producing 8-bit inverse. This image is then convolved with the coefficients $a_i$’s and generate the 1st level output ($y_1$ in the results).
5) In most of the cases, $y_1$ is good enough image for visual purposes. However, by incorporating an additional step of weighted sum of the two outputs to form a compound image ($y_{comp}$) can also help in improving the quality of the image

$$y_{comp} = \alpha \cdot y_1 + \beta \cdot u_1.$$  

(32)

6) $\alpha$ and $\beta$ are selected on the basis of relative projection of the images $y_1$ and $u_1$ onto the final compound image $y_{comp}$. The ad-hoc values that were used are $\alpha = 0.05$, and $\beta = 0.005$. The selection was based upon the visual quality of the re-constructed image. By fusing the two images in a weighted fashion, some of the texture details can be preserved from $y_1$ which are otherwise diffused in the reconstruction phase of $y_1$. The detailed study of the systematic influence of these parameters was not done in this work and will be performed in future.

B. Algorithm 2

The least-squares identification problem for the model in (30) can be posed as follows:

$$\min_{\theta} ||(y - \Phi \theta)||_2.$$  

(33)

It has been shown for 1-D systems in [11] that there exists a weight $W$ for which the weighted least-square problem

$$\min_{\theta} ||W(y - \Phi \theta)||_2$$  

(34)

which leads to the $H_{\infty}$ solution. The $H_{\infty}$ norm for 2-D system transfer function $F(z_1, z_2)$ is defined as

$$||F(z_1, z_2)||_{\infty} = \sup_{\omega_1} \sup_{\omega_2} |F(e^{j\omega_1}, e^{j\omega_2})|.$$  

(35)

The minimized $H_{\infty}$ norm is that of the frequency response error between the original image and the restored one. The weight $W$ is iteratively calculated as

$$W^{t+1} = d/c$$  

(36)

where $c$ is defined as

$$c = y - \Phi \theta$$  

(37)

c is a scaling factor whose value is taken as 0.01, and $I$ stands for $I$th iteration in the weighted least-square problem. At the next iteration, the weight in the weighted least-square problem is taken to be equal to this error. Therefore, the algorithm will decrease the error in the next iteration and get better solution. The experience from the 1-D case [7] shows that the algorithm converges to a near optimal solution after a few iterations. The identified model displayed significant improvement in performance over the conventional least squares. Note that the first step of the proposed algorithm gives the nonweighted least-squares solution. For processing convenience, instead of calculating huge inverses, the weights are calculated for each iteration as

$$u(n) = \left( \sum_{i=t-1}^{n-1} \lambda^{n-i} u(i) u(i)^T \right)^{-1} \sum_{i=t-1}^{n-1} \lambda^{n-i} u(i) y(i)$$  

(38)

or

$$u(n) = \xi(n)^{-1} \psi(n)$$  

(39)

where

$$\xi(n) = \sum_{i=t-1}^{n-1} \lambda^{n-i} u(i) u(i)^T$$  

(40)

$$\psi(n) = \sum_{i=t-1}^{n-1} \lambda^{n-i} u(i) y(i)$$  

and $\lambda$ represents the forgetting factor and its value has been taken as 0.97. The index starts from $i = 1$ since data samples prior to $i = 1$ are needed for the first estimate. The value of $i$ $1$ depends upon the order of the selected estimator and on the level of confidence in data samples. For recursive formulation, the information available at time $n - 1$

$$\xi(n) = \sum_{i=t-1}^{n} \lambda^{n-i} u(i) u(i)^T$$

$$= \lambda \sum_{i=t-1}^{n-1} \lambda^{n-1-i} u(i) u(i)^T + u(n) u(n)^T$$

$$= \lambda \xi(n-1) + u(n) u(n)^T$$

$$\psi(n) = \sum_{i=t-1}^{n} \lambda^{n-i} u(i) y(i)$$

$$= \lambda \sum_{i=t-1}^{n-1} \lambda^{n-1-i} u(i) y(i) + u(n) y(n)$$

$$= \lambda \psi(n-1) + u(n) y(n).$$  

(41)
Using the matrix inversion lemma

\[
\xi^{-1}(n) = \lambda^{-1}\xi^{-1}(n-1) - \frac{\lambda^{-2}\xi^{-1}(n-1)u(n)u^T(n)\xi^{-1}(n-1)}{1 + \lambda^{-1}u^T(n)\xi^{-1}(n-1)u(n)}. \tag{42}
\]

Therefore

\[
w(n) = [\xi(n)]^{-1}\psi(n) \\
= w(n-1) + [\xi(n)]^{-1}u(n)[y(n) - u(n)^Tw(n-1)] \\
= w(n-1) + k(n)\alpha(n). \tag{43}
\]

It has been shown for 1-D systems in [11] that the updated weights with respect to iteration count the weighted least-squares problem leads to the \(H_\infty\) solution. The algorithm’s outline is as follows.

1) **Initialization:**
   This part is essentially the same as the one in Algorithm 1 except for the calculation of coefficients is done through the weighted least-squares approach.

2) **Real-Time Application (Reconstruction):**
   This part is exactly the same as in Algorithm 1.

V. RESULTS

This section summarizes the initial results from the above algorithms. Four types of images have been used in this section, as shown in Figs. 6 and 7. Hand and face images were taken using IRPRO camera with 50 mm lens, while the distant surveillance scenes were taken using IRPRO with 75-mm lens.

- Human hand image: Both CCD and IR images.
- Distant scene image: Both CCD and IR images.
- Distant scene image with human subject.

The first two pair of images [Fig. 6(a)–(d)] are used to represent the situation in which the surveillance camera is used for near objects, while the third pair of images is used to simulate the actual surveillance situation where the objects are quite far. The image in Fig. 7 does not have the CCD counterpart, and, hence, it was used to create the actual scenario in which the background remains the same and another object is now present in the scene. Using the \(\hat{\theta}\) from the other two images of the scene, this new image is then enhanced. Since for surveillance applications, the environment does not change a lot, this step has to be done only once. Once the coefficients are known, then each successive frame is passed through the filter constituted by these coefficients to obtain an enhanced version of the input images.

A. Results from Algorithm 1

The identified or estimated coefficients for each model are structured in a column vector \(\hat{\theta}\) as

\[
\hat{\theta} = [a_1 \ a_2 \ a_3 \ a_4 \ a_5 \ a_6 \ a_7 \ b_{11} \ b_{12} \ b_{13} \ b_{14}]^T. \tag{44}
\]

Calculated \(\hat{\theta}\) for Algorithm 1

\[
\hat{\theta} = [-0.4813 - 0.1237 - 0.0717 - 0.0525 - 0.0502 - 0.0450 0.5646 1.0161 \\
-0.5609 \ 0.3694 \ 0.2172]^T \text{ (hand)} \tag{11}
\]

\[
\hat{\theta} = [-0.6716 - 0.1512 - 0.1123 - 0.1875 - 0.2352 - 0.0463 \ 0.7208 \ 0.2272 \\
0.0630 \ 0.2167 \ 0.1026]^T \text{ (face)} \tag{12}
\]

\[
\hat{\theta} = [-0.3414 \ 0.1115 - 0.0501 - 0.0546 \ 0.0057 - 0.0302 - 0.7091 - 0.3753 - 0.1059 \ 0.0587 \\
-0.6233]^T \text{ (Distant scene)} \tag{13}
\]

The results of application of this algorithm on Fig. 6(a)–(d) are shown in Fig. 8(c) and (d).
B. Results From Algorithm 2

Calculated $\hat{\theta}$ for Algorithm 2

$$\hat{\theta} = \begin{bmatrix} -0.7291 & 0.0038 & -0.2456 & -0.0195 \\ -0.1485 & 0.0813 & -0.0393 & 0.0911 & -0.2259 \\ -0.0445 & 0.1508 \end{bmatrix}^T \quad \text{(hand1)}$$

$$\hat{\theta} = \begin{bmatrix} -1.5540 & 0.4315 & 0.3791 & -0.1778 \\ -0.0021 & -0.0998 & 0.0232 & 0.1560 \\ -0.1862 & 0.1014 & -0.0672 \end{bmatrix}^T \quad \text{(face1)}$$

$$\hat{\theta} = \begin{bmatrix} -0.9919 & -0.0059 & -0.0855 & 0.2265 \\ -0.0097 & -0.1004 & -0.0326 & 0.0852 \\ -0.0750 & 0.1430 & -0.1205 \end{bmatrix}^T \quad \text{(Distant scene)}.$$

The results of the application of this algorithm on Fig. 6(a)–(d) are shown in Figs. 8(e) and (f). Similar results for Fig. 6(e) and (f) are shown in Fig. 9. Figs. 8 and 9 are sort of confirmatory results of the algorithms but only for the initialization phase. This is to imply here that in order to check if the algorithm is working precisely as expected then these images must depict a very sound reconstruction and image quality. As can be seen for these figures, the results are quite similar to the actual images and, therefore, conforms to the expected reconstruction. The following sections present the results of utilizing the same model of the background image with different real images as obtained by the surveillance camera. Also, since the surveillance is our main focus in this paper, only the distant scene coefficients are used with various surveillance scenarios encompassing different types of images as well as different artifact positions, times of the surveillance capture, and relationship between the foreground artifacts and background objects with respect to overlapping situations, i.e., if the object is front of a significant background object and/or behind a similar background object.

C. Reconstructing Images for Algorithm 1

Using the procedure outlined in Section IV-A2, the results obtained are shown in Fig. 10. Further to the images of Figs. 10 and 11, several other frames of the surveillance video were tested with the presented algorithms. Some of these are shown in Fig. 12.

Fig. 12 shows the compounded outputs only from the surveillance videos as they are subjected to the two presented algorithms. Various cases of varying object (person) in the image were tested to see the effects of background obscuring. Specifically, the change of a large object of the background can have a variety of effects including convolution with wrong model parameters, confusing the background with foreground artifact, and blurred reconstruction, etc. Since the reconstruction process is essentially a localized filtering operation with filter parameters being the coefficients of the estimated model, at times, the less prominent background gets a diffused look. Recalling the fact that the fundamental model utilized by both the algorithm has averaging involved in it since it is an ARMA system. Does this imply a handicap of the technique? The answer could be no or yes, depending upon the particular application in mind. Since the main target is surveillance applications, the relatively static background can be considered to be of less importance than some other artifact in foreground. Such an artifact would result in a slightly poor reconstruction, especially at its boundaries in the premodeled background. This is quite evident from the whitish blooming effect seen at the artifact boundary. However, the main background is still preserved to a large extent, even in the presence of the significant obscuring happening due to the artifact as in Fig. 12(e) and (f). Fig. 12(g) and (h) represents another interesting situation where the artifact is behind
Fig. 10. Results of reconstruction phase Surveillance image using Algorithm 1. (a) Reconstructed output image \( y \). (b) Reconstructed input image \( u \). (c) Compound image.

Fig. 11. Results of reconstruction phase Surveillance image using Algorithm 2: (a) Reconstructed output image \( y \). (b) Reconstructed input image \( u \). (c) Compound image.

Fig. 12. Various scenes from the surveillance videos. Left-hand column has the compounded output results for Algorithm 1, and the right-hand column shows the same for Algorithm 2.

a background object. Obviously, the hidden portion was completely missed while the remaining revealed portions are treated as foreground artifacts.

D. Comparison

A comparison of the proposed algorithms was made based on the visual quality of the results and the processing time needed for real time applications. These results were compared with a least-square estimator [4] and an \( H_\infty \) identifier [6] as benchmark algorithms.

- From the visual observations, the logical order of the superiority of the algorithms would be as follows.
  1) Algorithm 1.
  2) Algorithm 2.
  3) \( H_\infty \) identifier.
- Least-Square Estimator.
- In terms of initial processing time for the calculation of \( \theta \) from the IR and optical images, the ordered list is as follows.
  1) Least-square estimator (87 s).
  2) Algorithm 1 (159 s).
  3) \( H_\infty \) identifier (176 s).
  4) Algorithm 2 (203 s).
- However, the initialization phase is done offline and the numbers could be misleading since the accuracy that is brought in by the two proposed algorithms is at an increased processing cost. But since this is done in the beginning or only whenever the background changes (not so
common for still IR cameras). A better processing time measure can be utilized in terms of reconstruction speed and it is as follows.

1) Least-square estimator (16 ms).
2) Algorithm 1 (30 ms).
3) Algorithm 2 (35 ms).
4) $H_{\infty}$ identifier (45 ms).

Since these timings are calculated in the programs written in MATLAB 6.5.1, a significant reduction in the timing requirements is expected when implemented at the hardware level with fully dedicated processor. The usual frame rate is 30 frames per second for the IR camera’s video transmission which implies a processing time of $<33$ ms, and it can be safely expected that the algorithms would do just fine under these timing constraints.

With the above hierarchies in mind, the final implementation of the algorithms in hardware format will be completed in the next phase of this work. For Algorithm 2, the number of iterations is also an important parameter. It takes about 35 ms for three iterations. Again, a time restriction was imposed due to the real-time application scenario, and, hence, no more than three iterations could be implemented for real-time application.

VI. CONCLUSION

In this paper, some major findings have been presented. These are summarized as follows.

- Since classical $H_{\infty}$ is too complicated and time-expensive for real-time applications, alternative approaches were sought.
- As such, variants of LMS and WLS algorithms have been found to perform near $H_{\infty}$ bounds.
- Two new algorithms have been developed in this work using these variants and have been tested with several IR images.

One major contribution of this work is the idea that in order to improve the visual quality of IR images, their pixel space could be mapped to a corresponding optical pixel space. This was made possible by simultaneous image acquisition from IR and CCD cameras for a particular scene. This, of course, is done in the presence of ample amount of light and only once in the beginning. However, once the transformation is obtained, the actual image is then applied upon with these transformations to obtain an enhanced version.

ACKNOWLEDGMENT

The author would like to thank the reviewers whose keen reviews and critical comments have made this paper what it is now.

REFERENCES
