

Scene Based Non-Uniformity Correction in Thermal Images using Kalman Filter

Amir Averbuch¹ Gabi Liron¹ Ben Zion Bobrovsky²

¹School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel

²Department of Electrical Engineering, Faculty of Engineering, Tel Aviv University
Tel Aviv 69978, Israel

Abstract

Thermal array detectors, also known as focal plane arrays (FPA), are a rapidly developing technology and are used in a variety of civil, medical and military applications. The detectors, which are sensitive to radiation in the infrared band, output a high resolution low noise thermal picture. The existence of non-uniformities in the responsivity of the element array is a severe problem typical to FPA. These non-uniformities result in a fixed pattern “curtain” like feature that appear in the image. One of the most common methods to correct non-uniformity is the use of a uniform reference target. This type of non-uniformity correction has a number of disadvantages. The work presented in this paper proposes a new method to calibrate a thermal detector. The proposed correction method is scene based, relying only on the camera’s captured video sequence. The algorithm utilizes redundant information achieved from the thermal camera’s high sample rate, combined with the camera’s motion. The proposed correction algorithm contains two steps: 1. Application of frame registration that compensates for the camera motion. The registration process matches two consecutive frames and produces a residual (difference) frame. 2. Definition of the calibration parameters as a system of linear equations that is solved with the use of a Kalman filter. The Kalman filter tracks the value of each element specific responsivity value (unknown) through time. The main matrix is sparse. Extensive experimental work that demonstrates the superiority of this method is given. The algorithm necessitates modest computational power. It ran on a PC with a Pentium III 550 MHz processor.

1 Introduction

Thermal array detectors, also known as focal plane arrays (FPA), are a rapidly developing technology and are used in a variety of civil, medical and military applications. The detectors, which are sensitive to radiation in the infrared band (mostly $3 - 5\mu m$ and $8 - 12\mu m$), output a high resolution low noise thermal picture. Due to physical limitations, most FPA detectors need to be cooled down to extreme

low temperatures (less than 100 deg K). Naturally, this requirement has a dramatic effect on their price and their availability. Another severe problem typical to FPAs is the existence of non-uniformities in the responsivity of the element array. These sensitivity variations result in different outputs from elements receiving similar input flux (see Fig. 1).

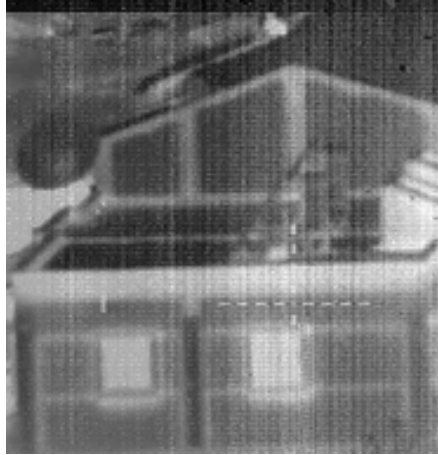


Figure 1: Typical example of a thermal picture with spatial noise.

In the more general sense, responsivity variations mean that a single mapping function (for all elements in the detector) between element input and output does not exist but rather every element or group of elements has its own individual and unknown map. As a result, when we watch a scene it is hard for the viewer to differentiate whether a specific picture pixel value is attributed to the scene (energy flux) or to the element's unique responsivity function. The differences in the element responsivity functions are generally referred to as non-uniformity. Unlike other imaging technologies, the non-uniformities in most FPA detector technologies can not be calibrated during manufacture because of drifts in the non-uniformity values. The calibration values change with time and between consecutive operations. As a result most detectors need to be calibrated on a regular basis. The time period between calibrations varies from a few minutes to a few hours [1].

Another related problem of FPA performance is the correction of bad elements, bad rows and columns. Bad elements, rows and columns are defined as picture objects, which have either very poor SNR or output a fixed signal. The phenomenon of bad elements is stable for long periods of time (unlike the drift in the non-uniformity values). Furthermore, bad elements tend to be permanent (a bad element will not become good). It is sometimes possible to recognize and fix bad elements during manufacture. Still, for most detectors the ability to recognize a bad element during the non-uniformity calibration is a powerful feature. The correction of previously detected bad elements is done in real time by interpolating the elements nearest neighbors [2].

The most common solution to perform non-uniformity correction (NUC) is the use of a uniform reference target. This target, located either inside the camera or external to it, supplies a known

input that can be used to evaluate the outputs of the detector's elements. Since the reference target is known to be uniform, a non-uniform output can be attributed to the elements non-uniformity. The correction of the elements non-uniformities can then be performed using various statistical methods. This NUC method has a number of disadvantages:

1. The camera has to contain a uniform reference target. Most internal targets are expensive electronic devices that take up space inside the camera and lower camera reliability. If the target is external some mechanical procedure is required to place the target before the detector.
2. During the calibration process the reference target blocks the detector's field of view. During calibration the camera is not functional.
3. The calibration is sensitive to changes in the input flux. This means that changes in the scene might dictate a re-calibration. For example, a camera that was calibrated indoors where the temperature is a comfortable 24 deg C will most likely need to be re-calibrated once taken outside where the temperature is 5 deg C .
4. The calibration quality is only as good as the uniformity of the reference target.

The work presented in this paper proposes a new method to calibrate the thermal detector. The proposed correction method uses a video sequence instead of using a uniform reference target. At the base of the proposed correction algorithm is the idea that redundant samples in the scene can be used to calculate the detector's non-uniformity. The redundant information is achieved from the thermal cameras high sample rate (30-60Hz), combined with utilization of camera motion. The combination of the two ensures multiple samples of the same location in the scene from different detector elements. The proposed correction algorithm contains two steps:

1. Application of frame registration to compensate for the camera motion. The registration process matches two consecutive frames and produces a residual (difference) frame. Each pixel in the difference frame contains the reading from an element (x, y) at time t and the reading of another element $(x - k_x, y - k_y)$ at time $t - 1$. Both elements (x, y) and $(x - k_x, y - k_y)$ are looking at the exact same location in the scene. A non-zero difference between them can be attributed to changes in the scene or to element responsitivity differences.
2. A system of linear equations is defined using the registration warp and the residual frame. The system has one unknown for every element of the array (76,800 unknowns for a 320×240 element detector). The system of equations is solved by using Kalman filter. The Kalman filter tracks the value of each elements specific responsitivity (unknown) through time. To accomplish this task the filter is applied iteratively on the video sequence. The Kalman filter produces a least

squares (LS) estimation of the non-uniformity. The governing matrices of the Kalman filter are sparse. Therefore, the implementation of the Kalman filter makes use of sparse matrix library and conjugate gradient for internal computations. A side benefit of the Kalman based correction process is the detection of bad elements that can be used for bad element correction. For an example of some experimental results of the correction algorithm see Figs. 2 and 3.

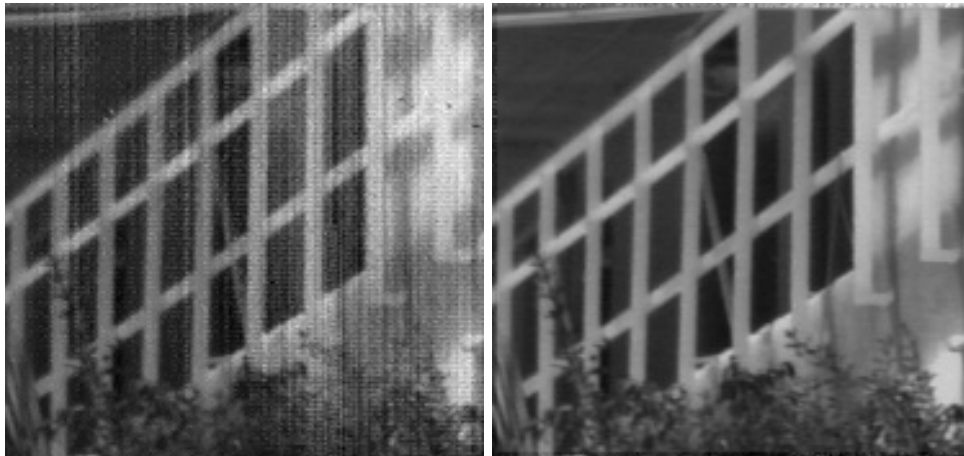


Figure 2: Left: An image from the original non-uniform sequence. The non-uniformity patterns are clearly visible. Right: The same image after the application of the Kalman filter based correction algorithm. The corrected image was produced by the correction algorithm after 171 iterations (at 30Hz, less than 6 seconds) with no prior knowledge of the scene or the nature of the non-uniformity.

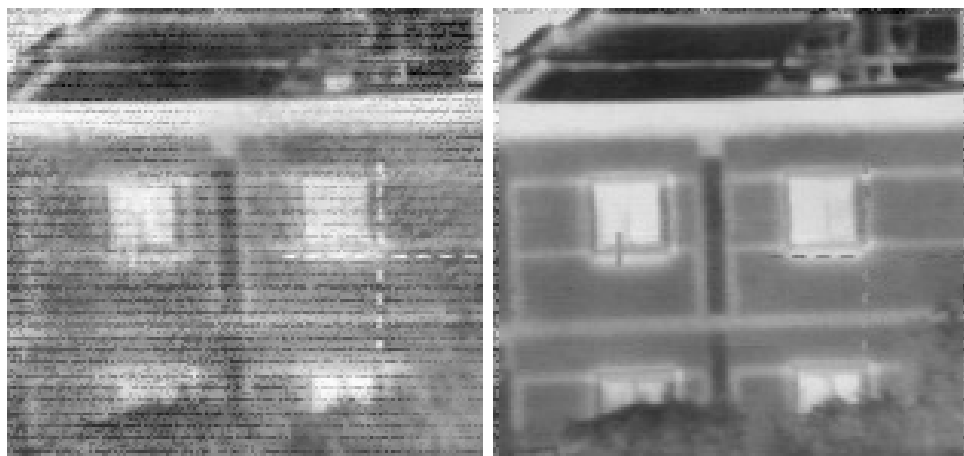


Figure 3: Left: An image from the original non-uniform sequence. Right: The same image after the application of the Kalman filter based correction algorithm. The corrected image was produced by the correction algorithm after 171 iterations with no prior knowledge of the scene or the nature of the non-uniformity.

The paper has the following structure: Section 2 contains a short introduction to thermal imaging

and to aspects of non-uniformity from an image processing point of view. Section 3 will introduce the state-of-the-art knowledge in non-uniformity correction and other related fields that can be useful to solve a scene based non-uniformity correction problem. Section 4 gives a detailed description of the proposed solution and section 5 concludes with the experimental results and a detailed evaluation of the performance of the algorithm.

2 Focal plane arrays: image processing point of view

In this section we give some background on the source of the noise that appears in the images.

2.1 Temporal and spatial noise

In [3] a thermal detectors behavior is characterized by a four dimensional space consisting of the following components: 1. Two dimensional location of every element in the array. 2, Signal transient in time of every element. 3. Responsivity curve of every element. Using this 4-D space the detectors temporal and spatial noise properties are defined.

2.1.1 Definition of temporal noise

The detectors temporal noise with background temperature τ , denoted by $\sigma^{temporal}(\tau)$, is defined as the mean of its elements temporal noise. Element temporal noise $\sigma_{x,y}^{temporal}(\tau)$, is calculated using the signal transient in time. The measured temporal noise of the element is influenced by the reference target temperature τ and the measurement duration time L . Assume $v_{x,y}^f(\tau, t)$ is the element's (x, y) output at time t given target temperature τ , the detector temporal noise is defined as ([3, 1]):

$$\begin{aligned}\sigma^{temporal}(\tau) &= \sqrt{\frac{\sum_{i=1}^N \sum_{j=1}^M [\sigma_{i,j}^{temporal}(\tau)]^2}{N \cdot M}} \\ \sigma_{i,j}^{temporal}(\tau) &= \sqrt{\frac{\sum_{t=1}^L [v_{i,j}^f(\tau, t) - \bar{v}_{i,j}(\tau)]^2}{L-1}} \\ \bar{v}_{i,j} &= \frac{\sum_{t=1}^L v_{i,j}^f(\tau, t)}{L}\end{aligned}$$

where N and M are the detector's length and width, respectively, and L is the number of frames in time that are used for the calculation. By using the derivative $\frac{\partial v}{\partial \tau}$ the detectors temporal noise can be translated from camera specific units (volts or grayscale) to degrees (deg C or deg K). The detector's temporal noise translated into temperature units is generally referred to as "noise equivalent temperature difference" (NETD). The advantage of the NETD measure is that it enables simple comparisons between different cameras.

2.1.2 Definition of spatial noise

The detectors spatial noise (spatial non-uniformity) is caused by the detectors elements responsivity curve component of the 4-D space [3]. When two elements have non-identical responsivity curves a similar input flux at a dissimilarity point will result in a dissimilar output. Similarly, a detector that contains elements with nonidentical responsivity will have a non-uniform output when exposed to a uniform input flux (uniform target). The effects of the non-uniform responsivity will also be viewable when the target is not uniform. The process of calibrating the detector's elements responsivity curve is known as non-uniformity correction (NUC). The non-uniformity correction process includes two steps. Initially, the average responsivity of the detector is determined and then every element's responsivity curve is adjusted to fit the average curve. To avoid effects of temporal noise in the NUC process the elements output values should be averaged (in time) of multiple frames (samples). As in the case of temporal noise, the target temperature τ influences the spatial noise. The spatial noise or non-uniformity is calculated using the following equations ([1, 3]):

$$\begin{aligned}\sigma^{spatial}(\tau) &= \sqrt{\frac{\sum_{i=1}^N \sum_{j=1}^M [\overline{v_{i,j}}(\tau) - \overline{v}(\tau)]^2}{N \cdot M - 1}} \\ \overline{v}(\tau) &= \frac{\sum_{t=1}^L \sum_{i=1}^N \sum_{j=2}^M v_{i,j}^f(\tau, t)}{L \cdot N \cdot M}\end{aligned}$$

The above definition of the spatial noise does not take into consideration different spatial frequencies of the spatial noise. The frequency of the noise is important because the eyes are sensitive different spatial frequencies. The sensitivity to low spatial frequency noise is smaller than that of high frequency noise. To complete the $\sigma^{spatial}$ measure, a secondary measure σ^{local} can be used. The new measure calculates only local non-uniformity, disregarding low spatial frequency noise (global noise). To calculate σ^{local} the detector area is divided into $K \times K$ elements sub-windows and the spatial noise of each sub-window is calculated. The value of σ^{local} is set as the mean of the spatial noise of the sub-windows.

2.2 Correction models for spatial non-uniformity

The non-uniformity correction models that are in use are based on polynomial equations [1, 3]. The polynomial degree of the correction model depends on the stability of the FPA and the complexity of the attached electronics. The simplest correction model, called "one point correction" (OPC), makes use of a zero degree polynomial. According to this model element non-uniformity is corrected using an offset value only. The optimal offset value is defined by the difference between the detectors mean output and the element's output at one temperature (hence its name). The calculated offset is stored in memory and subtracted from the elements output at every frame. The biggest advantage of this correction model is that the correction values are calculated only in one temperature. This simplifies

the electronics and shortens the calibration time. The disadvantage of this model is that it is less stable than higher degree polynomials. The reason for this is that the physics of the non-uniformity does not behave according to a zero degree polynomial (the true non-uniformity has non-additive components). As a result, when the target temperature deviates from the calibration temperature the offset value is no longer the ideal offset and a new NUC operation is needed. The correction offset value is computed using

$$o_{x,y} = v_{x,y}^f - \bar{v}^f \quad (2.1)$$

where $o_{x,y}$ is the element's (x, y) correction offset value, $v_{x,y}^f$ is the output of the element (x, y) of the detector, and \bar{v}^f is the mean output value of all the elements in the detector. Assuming the correction offset $o_{x,y}$ is known, the corrected element output value $v_{x,y}^c$ is determined by

$$v_{x,y}^c = v_{x,y}^f - o_{x,y} + n_{x,y} \quad (2.2)$$

where $n_{x,y}$ is an additive temporal noise.

Another very common correction model called “two point correction uses the difference between the element's response and the mean response at two different temperatures in order to estimate a first-degree polynomial [1, 3]. The first degree equation is defined by two parameters: gain $g_{x,y}$ and offset $o_{x,y}$ that are calculated with the following equation:

$$g_{x,y} = \frac{\bar{v}(\tau_{high}) - \bar{v}(\tau_{low})}{v_{x,y}^f(\tau_{high}) - v_{x,y}^f(\tau_{low})}, \quad o_{x,y} = g_{x,y} \cdot v_{x,y}^f(\tau_{low}) - \bar{v}(\tau_{low})$$

The gain and offset parameters are used to correct every elements output using: $v_{x,y}^c = g_{x,y} \cdot v_{x,y}^f - o_{x,y} + n_{x,y}$. The two temperatures τ_{low} and τ_{high} are usually chosen to be below and above the background temperature.

More complex models based upon higher degree polynomials exist but are less common than one and two-point correction [1]. The underlying idea of all models is that every detector element is sampled at a number of working temperatures and a polynomial is fitted to those points. The more points sampled the better the polynomial fit. A good fit results in a lower residual non-uniformity as the target temperature furthers from the calibration temperature. The only drawback of higher degree models is that they require more sampling points and are more sensitive to sampling error.

3 Survey of scene based non-uniformity algorithms and super-resolution

3.1 Scene based non-uniformity correction algorithms

Non-uniformity correction algorithms can be divided into two major groups. The first group contains methods that use a uniform target for the correction process. These methods, use either the one

point or two point correction models. The second group contains a variety of scene based correction algorithms. These algorithms correct non-uniformity without the use of a uniform target. Instead, scene based algorithms rely on the information gathered from the scene during normal operation of the thermal camera. The open literature contains three different classes of scene based correction algorithms.

3.1.1 Uniform statistics methods

These methods, which were introduced in [4, 7, 8, 9, 10, 11], assume that given camera motion and ample time the statistical behavior of the input flux of all the detectors elements is uniform. Therefore, any non-uniformities in the statistical behavior of the element output is attributed to element non-uniformities. In [4] a temporal lowpass is utilized by averaging the detector output over a long time period. The detectors correction is achieved by subtracting the result after low-pass filtering from the detector output. In [7, 8, 9, 10, 11] a two-point correction model is used. The gain and offset parameters are linked to the statistical behavior of element output. In [8, 9] the gain and offset are defined by the element's temporal standard deviation and mean, respectively. Implementation complexity considerations drive the work in [9]. This uses L_1 norm instead of L_2 in computing the standard deviation. The work in [9] also addresses the time length needed for the statistical data collection. In [7] a more complex model is given where the statistical output properties of the gain and offset are established using non-iterative, equations.

The big advantage of this class of algorithms is that they are simple to implement and work well if the statistical uniformity assumption is true. The main drawback is that the above assumption is very restrictive. This assumption is satisfied in cases where either a great deal of motion exists (e.g. airborne camera) or the time available for the statistical data collection is very long. In the general case the uniformity assumption is not guarantee (e.g. a land-based camera scanning the skyline horizontally with the top half containing sky and the bottom half containing land) and therefore the algorithms will fail.

3.1.2 Neural network approach

These methods, described in [4, 5, 11], use the two-point correction model as a base for a two layer neural networks. The networks training feedback uses the average of every four elements closest neighbors. This feedback estimates the desired corrected output of the element. The iterative update of the elements gain and offset parameters are then computed using steepest decent.

3.1.3 Micro-scanning devices

Two patents, which were described in [6, 12], suggested to correct detector non-uniformity by oversampling. The oversampling is achieved using scanning optics. In [6] an optical scanning mechanism produces multiple images that contain relative translated motion. The translation between the images is of known size (set by the mechanical mechanism). The 2-D optical translation is compensated by applying an inverse translation. As a result of this inverse translation all the frames are co-aligned. The second step of the correction averages the co-aligned frames. The application of the average operator improves the images SNR by a factor of the square root of the number of scanned frames.

The invention described in [12] used a predefined scan pattern to get multiple shifted frames. At the base of the correction algorithm is the idea that two elements watching the same location should have the same output, and that two elements traversing through the same route should have the same gradient. The scan pattern ensures that every scene location is sampled by four neighboring elements. Every elements gain factor is defined by the ratio between its gradient and the average gradient of its four nearest neighbors. After the gain is corrected, the offset is calculated as difference in between the element's output (gain corrected) and the average output of its four nearest neighbors.

3.2 Super-Resolution algorithms

Super resolution is a field of research that concentrates on utilizing redundant information found in continuous image sequences for image restoration. As such, this field is closely related to scene based non-uniformity correction. The main difference between these approaches is that super-resolution algorithms correct a known distortion operator (usually it is a superposition of blur, warp and downsampling operators) while scene based methods correct dynamic time and space varying linear distortions.

3.2.1 Back Projection

Back projection algorithms were introduced by Irani and Peleg in [13, 14, 15]. The algorithms define a set of operators (warp, blur, and downsampling) that were applied to create a set of low resolution frames from a theoretical high resolution scene.

3.2.2 Adaptive estimation

This algorithm, proposed in [17, 18], uses adaptive filters to restore blurred, decimated and noisy images. The low resolution sequence of the images was created by applying the same set of operators as in the back projection algorithm ([15]) on a theoretical high resolution image. The result of this derivation is the formation of a large system of linear equations. This system is solved iteratively by applying a Kalman filter. In [18] a number of iterative methods (steepest descent, and normalized

SD) are used in the computations of the filter.

4 Kalman filter based correction algorithm

The goal of this section is to introduce an algorithm that implements one point non-uniformity correction (OPC). The OPC was described in detail in section 2.2. The algorithm is scene based and utilizes Kalman filter methodology. The algorithm operates on a sequence of video images produced by a focal plane array (FPA) thermal camera (source raw images). The algorithm produces the same sequence (without delay) where the images are corrected to be with minimal additive non-uniformity patterns. The correction process does not utilize any filters (such as low-pass) that might change the detector's frequency transfer function and therefore the detector's effective resolution and features. The proposed method consumes $O(n)$ memory and $O(n)$ operations. A real time (RT) subset of the algorithm can be derived in it is out of the scope of this paper. In Fig. 4 we see the raw input image and the output after the application of the correction algorithm.

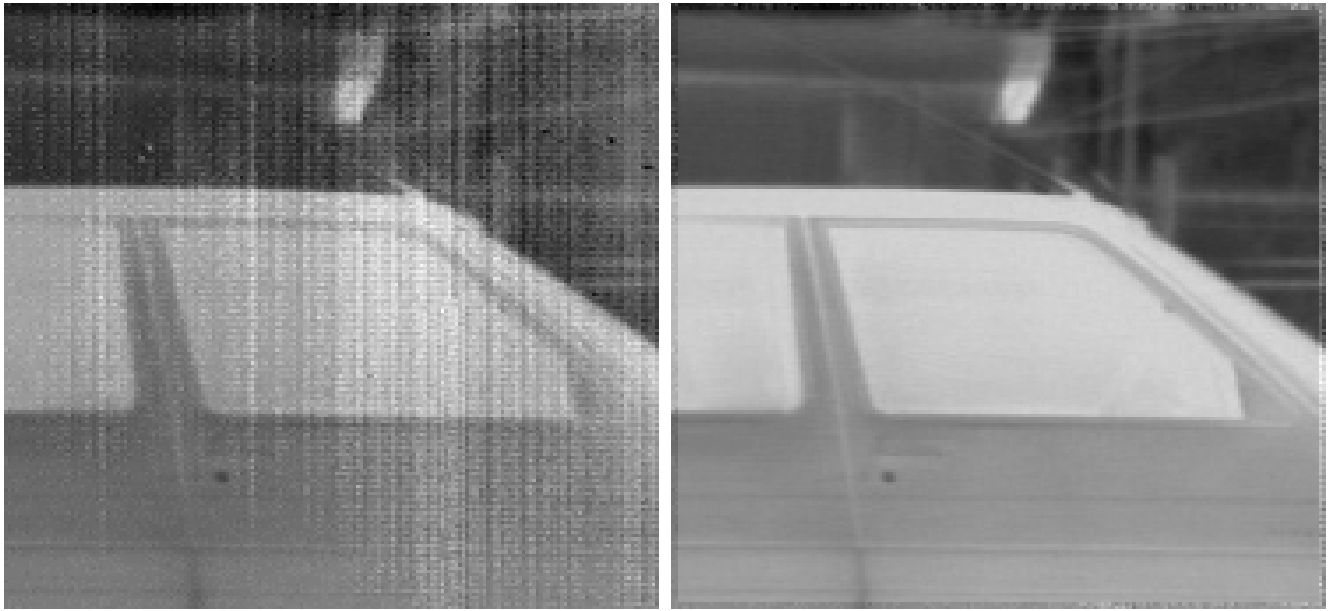


Figure 4: Left picture: The noisy image. Right picture: the image after the application of the correction algorithm which is explained in this section.

4.1 Prerequisites

Before we describe the correction algorithm two main assumptions are made about the correction model. These assumptions are considered to be prerequisites for the proper operation of the correction algorithm and its solution.

4.1.1 Algorithm Prerequisites

- (a) The scene being processed is static relative to the measurement frequency. Any changes in the scene between consecutive video frames (that are not due to camera motion) are small. Although this seems restrictive it is almost always true because the camera operating frequencies are 30-60Hz.
- (b) The detector element's non-uniformity offset value is nearly fixed throughout the correction phase. Changes in the offset values are allowed but should be small during the algorithm's execution. Since the correction time of the algorithm is a few seconds this is an unlimiting and realistic assumption.
- (c) Some camera motion is present. The extent of the motion between consecutive frames can vary from sub-pixel to a few pixels.

4.1.2 Solution prerequisites

- (d) We require that the correction algorithm is optimal at any given time. This algorithm, which is scene based, has an advantage over conventional OPC methods. Conventional methods are optimal only at a specific correction time. To ensure optimality through changes in the non-uniformity offsets we enable continuous changes in the offset values.
- (e) We require that the OPC will be continuous in time. This means that changes in the OPC between consecutive frames must be small. The reason for this requirement is that big changes in the OPC cause the view of the video sequence to blink. This requirement is not relevant to conventional OPC (not scene based) because it computes its correction offsets only once.

4.2 Correction algorithm

In order to simplify the description of the algorithm, we first correct a one dimensional detector (instead of the ultimate 2-D goal). Let us assume that the detector consists of a vector of N elements. Two readings were taken of the detector's elements at time $t - 1$ and t . Furthermore, it is known that in the time unit that passed between the two measurements the detector moved by k elements (see Fig. 5).

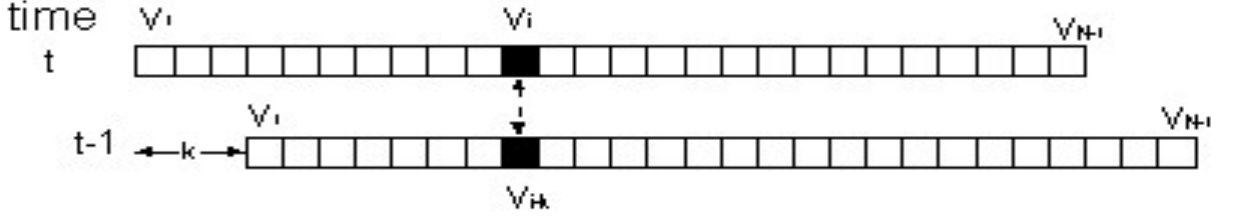


Figure 5: The camera moved k pixels to the left between time $t - 1$ and t . At time t pixel i is looking at the same spot in the scene pixel $i - k$ was at time $t - 1$.

Using the one-point correction model introduced in Eq. 2.2 the two measurements at times $t - 1$ and t can be described by:

$$\begin{aligned} v_{i-k}^c(t-1) &= v_{i-k}^f(t-1) - o_{i-k}(t-1) + n \\ v_i^c(t) &= v_i^f(t) - o_i(t) + n \end{aligned} \quad (4.1)$$

where $v_i^f(t)$ is the output of element i of the detector at time t , $v_i^c(t)$ is the “correct” value of element i , $o_i(t)$ is the i th element offset (which we want to find and correct), and n is an additive temporal noise with mean 0 and standard deviation σ . The value of σ is defined by the detector’s temporal noise parameter $\sigma^{temporal}$. The value of σ is determined during the detector’s manufacture or through statistics which was gathered during the detector’s operation. If we combine $v_i^c(t) \approx v_{i-k}^c(t-1)$, $i = 1, \dots, N$ (prerequisite a.) with Eq. (4.1) we get $o_i - o_{i-k} = v_i^f(t) - v_i^c(t) - [v_{i-k}^f(t-1) - v_{i-k}^c(t-1)] + n$ then

$$o_i - o_{i-k} = v_i^f(t) - v_{i-k}^f(t-1) + n \quad n = N(0, \sqrt{2}\sigma). \quad (4.2)$$

Equation (4.2) is at the very heart of the correction algorithm. It defines the connection between the non-uniformity offset \underline{o} and the difference frame obtained from registering two consecutive video frames. For the detector’s N elements we can expand Eq. (4.2) to a system of equations, :

$$\begin{aligned} o_1 - o_{1-k} &= v_1^f(t) - v_{1-k}^f(t-1) + n \\ &\vdots \\ o_N - o_{N-k} &= v_N^f(t) - v_{N-k}^f(t-1) + n \end{aligned} \quad (4.3)$$

Equation (4.3) is invariant to “global” offsets. This means that for a given solution \underline{o} it is possible to add a fixed offset to all o_i , $0 \leq i < N$ without changing the validity of the solution. This property is mentioned since the solution to Eq. (4.3) does not guarantee a unique correction. To ensure a unique OPC we either impose a zero average constraint on Eq. (4.3) or normalize the average of \underline{o} after it is found. The reason for Eq. (4.3) invariance to global additions is demonstrated in the following: Assume $o'_i = o_i + C \quad 0 \leq i < N$ then from $o_i - o_{i-k} = v_i^f(t) - v_{i-k}^f(t)$ we get $o'_i - o'_{i-k} = v_i^f(t) + C - [v_{i-k}^f(t) + C] = v_i^f(t) - v_{i-k}^f(t)$.

The system of equations (4.3) is not of full rank. This is because $k \neq 0$ causes $i - k < 1$ or $i - k > N$. In visualization terms, the system includes border elements in frame t that do not have partners (match) in frame $t - 1$ (see Fig. 6). In the 1-D case, a movement of k pixels causes the system to contain $N - k$ equations, where in the 2-D case a horizontal or vertical movement of k pixels causes k rows or columns to be left out.

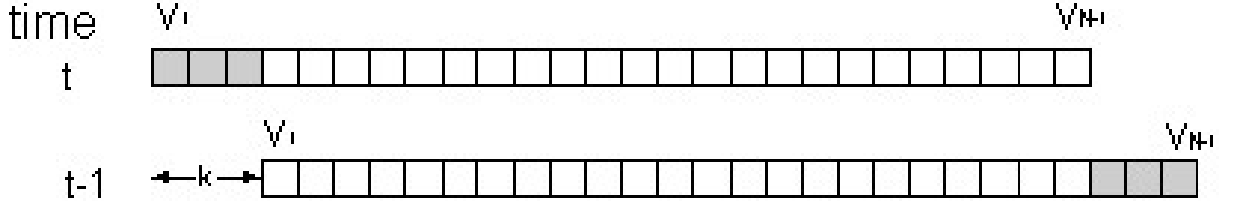


Figure 6: The camera moved k pixels to the left between time $t - 1$ and t . Some of the elements in time t do not have partner elements in time $t - 1$ (marked in gray).

As a result of the singularity of the system the solution of (4.3) is not unique. To guarantee a unique solution the stability of the offset, \underline{o} , is used (prerequisite b.) and the system is extended to contain equations produced over a number of frames. The new system of equations is of full rank with a unique solution.

$$\begin{aligned}
 \left. \begin{aligned}
 o_1 - o_{1-k(t,t-1)} &= v_1^f(t) - v_{1-k(t,t-1)}^f(t-1) + n \\
 &\vdots \\
 o_N - o_{N-k(t,t-1)} &= v_N^f(t) - v_{N-k(t,t-1)}^f(t-1) + n
 \end{aligned} \right\} N - k(t, t - 1) \\
 \left. \begin{aligned}
 o_1 - o_{1-k(t+1,t)} &= v_1^f(t+1) - v_{1-k(t+1,t)}^f(t) + n \\
 &\vdots \\
 o_N - o_{N-k(t+1,t)} &= v_N^f(t+1) - v_{N-k(t+1,t)}^f(t) + n
 \end{aligned} \right\} N - k(t + 1, t) \quad (4.4) \\
 &\vdots \\
 \left. \begin{aligned}
 o_1 - o_{1-k(t+l,t+l-1)} &= v_1^f(t+l) - v_{1-k(t+l,t+l-1)}^f(t+l-1) + \eta \\
 &\vdots \\
 o_N - o_{N-k(t+l,t+l-1)} &= v_N^f(t+l) - v_{N-k(t+l,t+l-1)}^f(t+l-1) + \eta
 \end{aligned} \right\} N - k(t + l, t + l - 1)
 \end{aligned}$$

The solution of Eq. (4.4) results in a better approximation of $\hat{\underline{o}}$ than that of (4.3) but its use is still not recommended due to the following drawbacks:

1. Eq. (4.4) uses at most l frames. The gathered data before time $t - l$ is not considered (contradictory to prerequisite d.). Naturally, the more frames used more accuracy is obtained. On the other hand, using many frames is computationally expensive.

2. The system is redefined every l frames and no continuity restrictions are applied between consecutive computation cycles of Eq. (4.4). This is contradictory to prerequisite e. Discontinuities in the OPC will result in a blinking video sequence.

4.3 Kalman Filter

Prerequisites d. and e. were used in section 4.2 to exclude the use of Eq. (4.4) for calculating the OPC. This exclusion resulted from the fact that conventional methods for solving Eq. (4.4) do not ensure continuity and do not maximize the use of the available data. One method to solve Eq. (4.3) expanded over a few frames that satisfy the dynamic requirements of both d. and e. is to use a Kalman filter. The Kalman filter tracks the non-uniformity in the video sequence.

4.3.1 Background

Kalman filter is an optimal linear adaptive (dynamic) estimator that is usually used in control problems and tracking. The “optimality” means that the filter is LS optimal. Adaptive / dynamic means that it enables the estimation to evolve through time as a result of changes in the inputs and according to prior knowledge. Kalman filters are adapt to work in noisy environments. Kalman filter requires the definition of two models:

- A system model that describes the expected evolution through time of the system variables.
- A measurement model that describes the connection between the input data recieved and the system variables.

The filter utilizes the two models in a two step approach. The first step is extrapolation and the second is update. The extrapolation step uses the system model to extrapolate the values of the system variables at time t given their values at time $t - 1$. The variable extrapolation does not use input data recieved at time t but only pior knowlegde of the system’s behavior. The update step uses the extrapolated variables at time t and the input data at time t to estimate a new variable value. The estimation is based on the measurement model.

4.3.2 Kalman filter definition

Before describing the equations for the filter’s two step model the following convention are needed. These conventions were adopted from [19].

- $\underline{q}_t(\pm)$ - System variables at time t . The $(-)$ marks the extrapolated value where $(+)$ is the estimated value.
- \underline{z}_t - The input data at time t (measurement vector).
- Φ_t - The state transition matrix defines the evolution of the system variables through time.
- H_t - The problem definition matrix that defines the connection between the system variables \underline{q}_t and the measurement \underline{z}_t .
- w_t - The system model noise. The system model noise defines the inaccuracy estimate for the the system model. It has normal distribution with zero mean and standard deviation Q_t .
- v_t - The measurement noise. It has normal distribution with zero mean and standard deviation R_t .

The Kalman filter is defined by the subsequent equations:

1. System and measurement models:

$$\begin{aligned}\hat{\underline{q}}_t &= \Phi_{t-1}\hat{\underline{q}}_{t-1} + w_{t-1}, \quad w_t = N(0, Q_t) \\ \underline{z}_t &= H_t\underline{q}_t + v_t, \quad v_t = N(0, R_t)\end{aligned}$$

2. Extrapolation step:

$$\begin{aligned}\hat{\underline{q}}_t(-) &= \Phi_{t-1}\hat{\underline{q}}_{t-1}(+) \\ P_t(-) &= \Phi_{t-1}P_{t-1}(+)\Phi_{t-1}^T + Q_{t-1}\end{aligned}\tag{4.5}$$

3. Update step:

$$\begin{aligned}K_t &= P_t(-)H_t^T[H_tP_t(-)H_t^T + R_t]^{-1} \\ P_t(+) &= [I - K_tH_t]P_t(-) \\ \hat{\underline{q}}_t(+) &= \hat{\underline{q}}_t(-) + K_t[\underline{z}_t - H_t\hat{\underline{q}}_t(-)]\end{aligned}\tag{4.6}$$

or

$$\begin{aligned}P_t^{-1}(+) &= P_t^{-1}(-) + H_t^T R_t^{-1} H_t \\ K_t &= P_t(+)\Phi_{t-1}^T R_t^{-1}\end{aligned}\tag{4.7}$$

The general operation of Kalman filter is illustrated by Fig. 7.

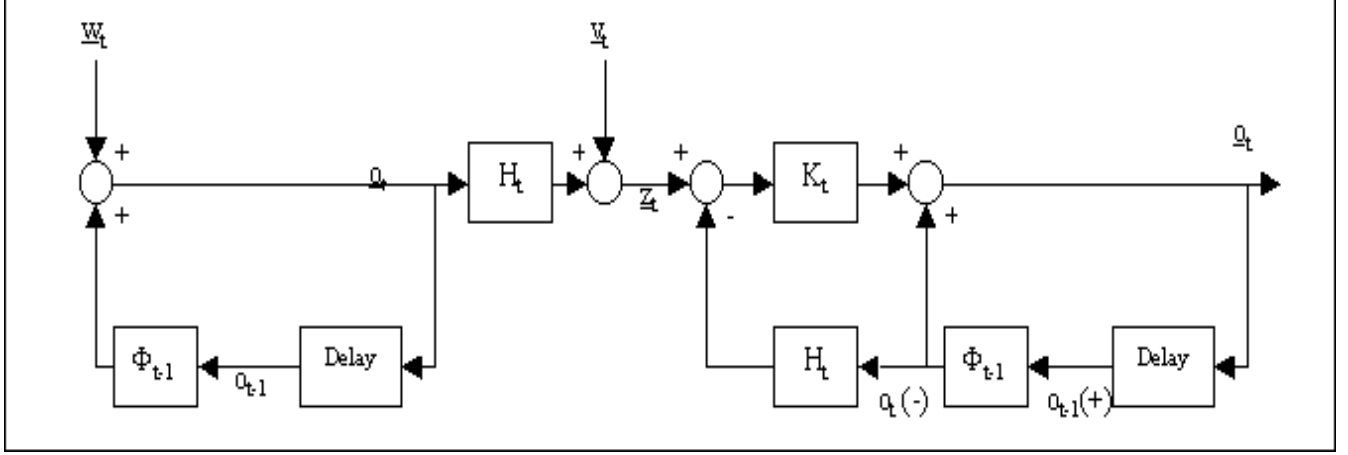


Figure 7: The general Kalman filter model.

4.4 Kalman filter implementation

In this section we describe how the Kalman filter addresses the specific OPC problem. The OPC problem is linear and dynamic, the measurements are noisy and we wish that the extracted solution to be optimal. Therefore, Kalman filter is the right tool. The properties of the Kalman filter ensure that both prerequisite d. (optimality) and e. (continuity) are obtained and preserved.

Equation (4.3) can be used as the measurement model of the Kalman filter. Rephrasing (4.3) according to Kalman filter terminology we get

$$\underline{z}_t = H_t \underline{o}_t + \underline{n} \quad (4.8)$$

where \underline{z}_t is the measurement data generated from the registration difference frame (between frame t and $t-1$), and H_t is the system problem matrix. The construction of \underline{z}_t and H_t is explained in section 4.5.

If we wish to solve Eq. (4.8) in a non-iterative least squares method we need to find \underline{o}_t that minimizes $e^2 = (\underline{z} - H\hat{o})^T (\underline{z} - H\hat{o})$. This is achieved by setting $\frac{\partial e^2}{\partial \hat{o}} = 0$, in other words, \hat{o}_t has to satisfy $\hat{o}_t = (H^T H)^{-1} H^T \underline{z}$.

To have a full Kalman filter description we will add the following system model

$$\hat{o}_t = I \hat{o}_{t-1} + W_{t-1} \quad (4.9)$$

to the measurement model of (4.8). The system model describes the evolution of \underline{o}_t between consecutive measurements according to a predefined behavior. In our case, \underline{o}_t is not expected to evolve (prerequisite b.). We shall therefore assume $\Phi_t = I$.

If we combine our specific system and measurement models Eqs. (4.8) and (4.9) with the general Kalman filter settings (described by Eqs.(4.5) - (4.7)) we get a simplified set of equations:

$$P_t(-) = P_{t-1}(+) + Q_{t-1} \quad (4.10)$$

$$P_t^{-1}(+) = P_t^{-1}(-) + H_t^T R_t^{-1} H_t, \quad R_t^{-1} = \frac{1}{\sqrt{2}\sigma} \quad (4.11)$$

$$K_t = P_t(+)\mathit{H}_t^T R_t^{-1} \quad (4.12)$$

$$\hat{\underline{q}}_t = \hat{\underline{q}}_{t-1} + K_t[\underline{z}_t - H_t\hat{\underline{q}}_{t-1}] \quad (4.13)$$

When we experimented with various values of Q_t (Q_t is the standard deviation value of w_t) it was found that using no system noise ($Q_t = 0$) gives good results. This is explained by the stability of \underline{q}_t . Another possible explanation is the use of finite precision calculations and approximation methods throughout the filter's implementation. The bottom line result is that the value of Q_t can be set to zero without decreasing solution accuracy or the dynamic properties of $\hat{\underline{q}}_t$. The specific instance of the Kalman filter with $\Phi = I$ and $Q_t = 0$ is an implementation of recursive least squares (RLS). The use of the Kalman filter instead of the RLS is done to preserve the generality of the algorithm.

4.5 Generating \underline{q} , \underline{z} and H

To generate the vector \underline{z}_t and the problem matrix H_t , the warp between every two consecutive video frames is needed. For now, let us assume that a "black box" registration module exists that produces the correct horizontal and vertical translation parameters. More in-depth information about registration considerations is given in section 4.10.

\underline{q}_t is the 1-D vector of unknown offsets. To create \underline{q}_t the 2-D detector array is scanned by rows and every row is concatenated at the end of the previous row. The result of this operation is a vector of length $N \times M$. Similarly, the vector \underline{z}_t is generated through the scan of the 2-D array that contains the registration difference frame (this is illustrated in Fig. 8). Assume

$$\begin{aligned} y &= (i \div N) & y' &= (i \div N - k_y) \\ x &= (i \% N) & x' &= (i \% N - k_x) \end{aligned}$$

then the definition of \underline{z}_t appears as

$$z_i = \begin{cases} v_{y,x}^f(t) - v_{y',x'}^f(t-1) & y' \in [0, \dots, N-1], \quad x' \in [0, \dots, M-1] \\ 0 & \text{otherwise} \end{cases} \quad (4.14)$$

for $i = 0, \dots, N \cdot M - 1$.

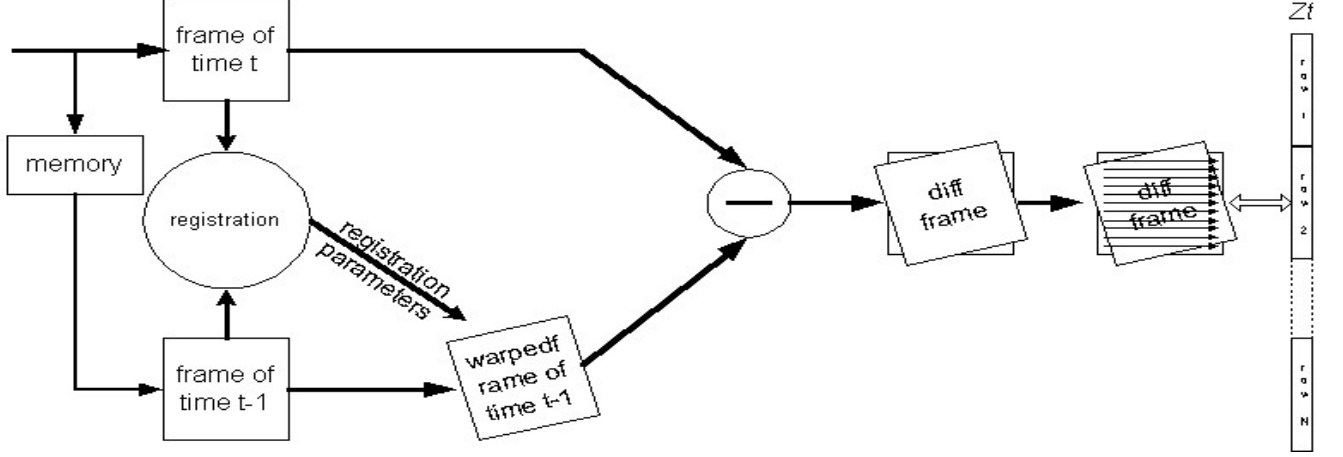


Figure 8: The creation process of vector \underline{z}_t .

The problem matrix H , which implements Eq.(4.3), is given by:

$$h_{i,j} = \begin{cases} 1 & i = j, \quad y' \in [0, \dots, N-1], \quad x' \in [0, \dots, M-1] \\ -1 & j = (y' \cdot N) + x', \quad y' \in [0, \dots, N-1], \quad x' \in [0, \dots, M-1] \\ 0 & \text{otherwise} \end{cases} \quad (4.15)$$

for $i, j = 0, \dots, N \cdot M - 1$.

The matrix H created according to Eq. (4.15) is a sparse $(N \cdot M) \times (N \cdot M)$ matrix. It contains up to two non-zero values per row. The matrix H is narrow banded with 1s on the main diagonal and -1 on a secondary diagonal that is horizontally shifted by a few columns.

Because the algorithm has to work with sub pixel registration an improvement is introduced to Eqs. (4.3) and (4.15) by using nearest neighbor interpolation. The interpolation is used when k_y or k_x or both are not integers. Instead of using o_{i-k} in (4.3) we use a linear combination of its four closest neighbors. The interpolated value of v_{i-k}^f is obtained from the registration module that registers the two frames with sub-pixel accuracy. Let \downarrow be the lowest integer part of the number, and \uparrow be the largest integer part of the number. Assume that

$$\alpha = k_y - \downarrow k_y$$

$$\beta = k_x - \downarrow k_x$$

then Eq. (4.3) is replaced with the following:

$$\begin{aligned} o_{y,x} - \left[\alpha \beta o_{\downarrow(y-k_y)\downarrow(x-k_x)} + (\alpha - 1) \beta o_{\uparrow(y-k_y)\downarrow(x-k_x)} + \alpha(\beta - 1) o_{\downarrow(y-k_y)\uparrow(x-k_x)} \right. \\ \left. + (\alpha - 1)(\beta - 1) o_{\uparrow(y-k_y)\uparrow(x-k_x)} \right] = R_{x,y}(t, t-1) + n \end{aligned} \quad (4.16)$$

where $R_{x,y}(t, t-1)$ is pixel (x, y) of the difference frame created by the sub-pixel registration between frames $t-1$ and t .

The construction rule for the matrix H , that supports sub-pixel registration is given by:

$$h_{i,j} = \begin{cases} 1 & i = j \\ -\alpha\beta & j = [\downarrow y' \cdot N] + \downarrow x' \\ -(1-\alpha)\beta & j = [\uparrow y' \cdot N] + \downarrow x' \\ -\alpha(1-\beta) & j = [\downarrow y' \cdot N] + \uparrow x' \\ -(1-\alpha)(1-\beta) & j = [\uparrow y' \cdot N] + \uparrow x' \\ 0 & \text{otherwise} \end{cases} \quad (4.17)$$

for all $y' \in [0, \dots, N-1], x' \in [0, \dots, M-1]$.

The modified H has at most five non-zero entries in each row and it is still classified as narrow banded, as demonstrated in Fig. 9.

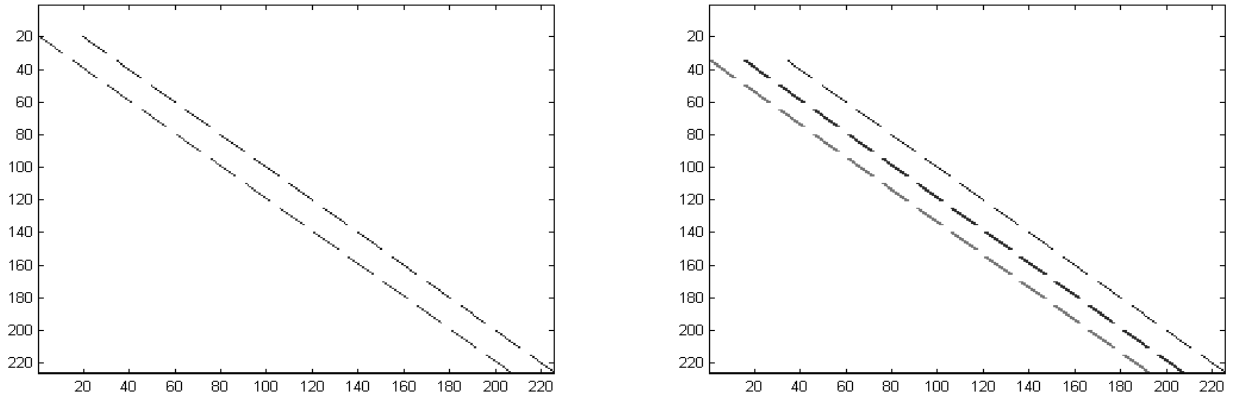


Figure 9: Two examples of the matrix H where the detector has 15×15 elements. The left picture is a result of a 1 vertical and 4 horizontal units translation. The right picture is a result of a 1.2 vertical and 3.5 horizontal units translation. Row i with the 2-D coordinate $\langle i \div N - k_y, i \% N - k_x \rangle$ which exceeds the detector boundaries is zero (blank).

4.6 The sparsity of the matrices

4.6.1 The matrix H

As defined by Eqs. (4.17) the problem matrix H contains at most 5 non zero elements per row. The matrix H is sparse and has a sparsity ratio (ratio between number of non-zero elements and overall elements in H) of $5/A$, where A is the detector area ($N \cdot M$). For example, a detector of a 100×100 elements will have an H matrix which contains less than 0.05% non-zero elements (0.002% for a solution that does not support sub-pixel registration).

4.6.2 The matrix P_t^{-1}

The inverse error covariance matrix P_t^{-1} , used in (4.11), is obtained by iteratively adding $H_t^T R_t^{-1} H_t$ to P_{t-1}^{-1} . Because R_t^{-1} is diagonal and given the structure of H and H^T the resulting P_t^{-1} will also be sparse. The number of non-zero elements in P^{-1} is bound by $3A(S_w^2)$, where S_w is the registration search window diameter. The sparsity ratio of P^{-1} is bounded by $3(S_w^2)/A$.

Proof:

1. R_t^{-1} is diagonal and therefore does not effect the location of non-zero elements.
2. According to Eq. (4.15) (for whole pixel registration) and the transpose operation definition, H_t^T contains at most 2 non-zero elements per row, i , at locations (i, i) and $(i, i + k)$ ($k = k_y \cdot N + k_x$ is the 1-D equivalent of the 2-D registration warp).
3. According to equation (4.15) H_t contains at most 2 non-zero elements per column j at locations (j, j) and $(j + k, j)$.
4. The covariance inverse update can have only 3 non-zero elements per row at location (i, i) , $(i, i - k)$, $(i, i + k)$. This is true since

$$\begin{aligned} \Delta P_{i,j}^{-1} \neq 0 &\Leftrightarrow (H_t^T R_t^{-1} H_t)_{i,j} \neq 0 \Leftrightarrow h_{i,i}^T \cdot h_{i,j} + h_{i,i+k}^T \cdot h_{i+k,j} \neq 0 \\ h_{i,i}^T \cdot h_{i,j} \neq 0 &\Leftrightarrow i = j \text{ or } i = j + k \\ h_{i,i+k}^T \cdot h_{i+k,j} \neq 0 &\Leftrightarrow i + k = j \text{ or } i + k = j + k. \end{aligned}$$

From this we derive

$$\Delta p_{i,j}^{-1} \neq 0 \implies i = j \text{ or } i = j - k \text{ or } i = j + k.$$

5. Since the number of possible k values is restricted by S_w^2 the maximum number of non-zero elements is bound by $3(S_w^2)$ per row.
6. The mMatrix H_t^T and H_t that are used with subpixel registration are restricted to have non-zero locations that are defined by the four nearest neighbours as appeared in Eq. (4.16).

Figure 10 illustrates the sparsity of the update matrix ΔP^{-1} and matrix P^{-1} . In the left figure the size of the matrix is $10,000 \times 10,000$ and the number of non-zero elements in one update cycle is 29,000. While in the right picture we have 406,780 non-zero elements after 50 iterations of the Kalman filter.

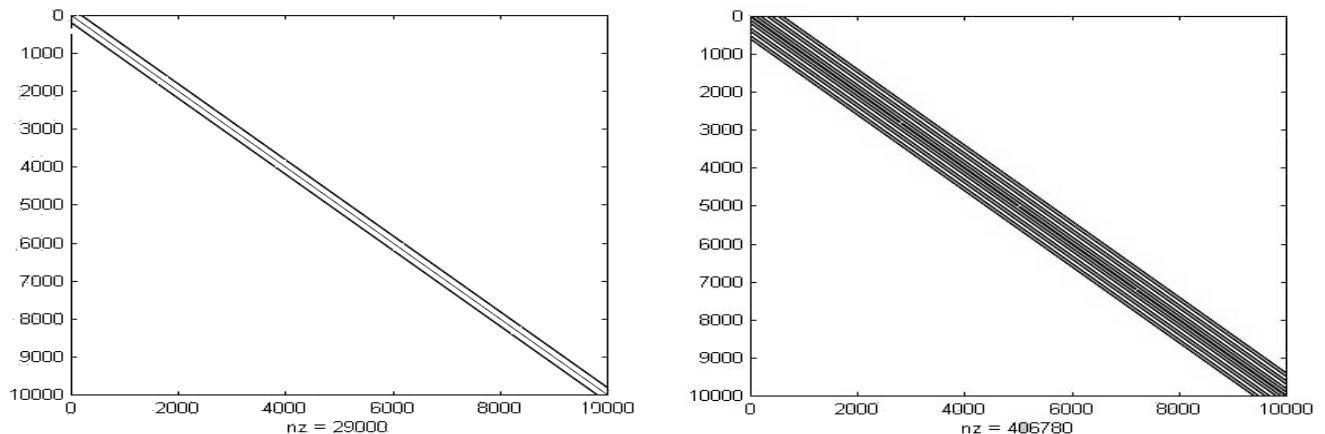


Figure 10: Left picture: Matrix ΔP^{-1} of a detector of size 100×100 and a registration translation of $(2, 3)$. Right picture: Matrix P^{-1} of a detector of size 100×100 created after 50 iterations of Kalman filter. P^{-1} has a sparsity ratio of about $4e - 3$. *nz* means number of non-zero elements.

4.6.3 The matrix P_t

Equation (4.12) requires the inversion of the sparse matrix P_t^{-1} . Because P is the covariance error matrix it is expected from the physics of the problem that P is sparse. The reason is that the mutual error variance of distant elements should be zero. But, since sparse matrix inversion does not guarantee sparsity the use of an inversion method can destroy the sparsity format. Furthermore, even if P is sparse it can not be guaranteed that temporary matrices used during the inversion of P^{-1} are all sparse. Computational inaccuracies can also reduce the sparsity of P by creating small non-zero elements.

We present two possible solutions for the implementation of the Kalman filter.

4.7 Solution I: Straight forward method

The straight forward method for solving the Kalman filter equations Eqs. (4.10)-(4.13) is to use the inverse operation on P_t^{-1} . The resulting matrix P_t is then used to calculate K_t and \hat{z}_t .

As a reminder let us first review the sizes of the vectors and matrices involved in the Kalman filter:

$$\begin{aligned}
 A &= N \cdot M \text{ (detector area in elements)} \\
 size(\underline{q}) &= A \times 1 \\
 size(\underline{z}) &= A \times 1 \\
 size(H) &= A \times A \\
 size(P) &= A \times A
 \end{aligned}$$

4.7.1 Memory requirements

The vector \underline{z} and the matrix H are recomputed in every iteration according to the registration results. Both can be computed on the fly and therefore require $O(1)$ memory. This does not include $O(A)$ memory needed for the registration module. The vector \underline{q} and the inverse covariance error matrix P^{-1} has to be stored during the entire run of the algorithm. The memory needed to store \underline{q} and P^{-1} is $O(A)$. The memory needed for the inverse of P^{-1} is $O(A^2)$.

4.7.2 Computational requirements

Because of the size of the involved matrices it is assumed that all the operations on sparse matrices are done utilizing the matrix sparsity. Implementing the Kalman filter as described in Eqs. (4.10)-(4.13) has the following complexity:

$P_t^{-1} = P_{t-1}^{-1} + H_t^T R_t^{-1} H_t$ – The indices of P_t^{-1} to be updated are known and their number is bounded by $3A$. The value of $H_t^T R_t^{-1} H_t$ for each of these indices can be calculated in $O(1)$ time because the location of the non-zero indices in H_t^T and H_t is known. Overall requirements to update P_t^{-1} is $O(A)$.

$K_t = P_t H_t^T R_t^{-1}$ – Inverting P_t^{-1} requires $O(A^3)$. Multiplying the non-sparse P_t with the sparse H_t^T requires $O(A^2)$.

$\underline{q}_t = \underline{q}_{t-1} + K_t(\underline{z}_t - H_t \underline{q}_{t-1})$ – Multiplying the non-sparse matrix K_t (of size $A \times A$) by a vector of length A requires $O(A^2)$ operations.

The overall computational requirements of one iteration of the filter assuming sparse implementation is therefore $O(A^3)$.

4.8 Solution II: Conjugate Gradient method (CG)

To better utilize the sparsity of P_t^{-1} we use an iterative method to avoid the need to invert the matrix P_t^{-1} . By using algebraic manipulation to restate Eqs. (4.11), (4.12) and by assuming $A_t \triangleq P_t^{-1}$, $b_t \triangleq H_t^T R_t^{-1} [\underline{z}_t - H_t \hat{\underline{q}}_{t-1}]$ we get

$$\begin{aligned} \underline{x}_t &= A^{-1} \underline{b}_t \\ \hat{\underline{q}}_t(+)&= \hat{\underline{q}}_{t-1} + \underline{x}_t \end{aligned} \tag{4.18}$$

Conjugate gradient [16] is iteratively applied to solve Eq. (4.18).

4.8.1 Memory requirements

As shown in 4.6 the matrix P_t^{-1} is sparse and requires $3S_w^2 A$ memory. The matrix P_t is not used in the computation. Applying conjugate gradients to solve $x_t = (P_t^{-1})^{-1} b_t$ requires $O(A)$ memory.

4.8.2 Computational requirements

Implementing the Kalman filter as described in Eqs. (4.10) and (4.18) requires the following operations:

$$P_t^{-1} = P_{t-1}^{-1} + H_t^T R_t^{-1} H_t - \text{ Same as in staight forward method.}$$

$$b_t = H_t^T R_t^{-1} (z_t - H_t \hat{o}_{t-1}) - \text{ Since } H \text{ and } H^T \text{ have at most 5 non-zero indices per row/column and their locations are known, multiplying the matrix by a vector takes only } O(A) \text{ operations.}$$

Application of the conjugate gradient to solve $x_t = (P_t^{-1})^{-1} b_t$ requires $O(A)$ time per iteration. Although theoretically the number of iterations is bounded by A , experimental results show that ten iterations (even five) result in a satisfactory accuracy of \hat{x}_t . If a constant is used to limit the number of iteration of the CG the overall computational requirements for this computation is $O(A)$. The overall complexity of a single iteration of the Kalman filter is therefore $O(A)$.

4.9 Bad elements detection

The detection of bad elements is based on the fact that a bad element does not behave according to (4.1), and therefore can not be solved correctly using (4.2). When we have a bad element i ($v_i^f(t) \approx C$), with a “good” registration partner (match) $i - k$ we recieve the following estimation error:

$$\begin{aligned} |\underline{z}_t - H_t \underline{o}_t|_i &= |z_i(t) - [o_i(t) - o_{i-k}(t-1)]| = \left| v_i^f(t) - v_{i-k}^f(t-1) - [o_i(t) - o_{i-k}(t-1)] \right| \\ \hat{o}_{i-k}(t-1) &\approx v_{i-k}^f(t-1) - v_{i-k}^c(t-1) \end{aligned}$$

and from these we get

$$|\underline{z}_t - H_t \underline{o}_t|_i \approx |o_i(t) + v_{i-k}^c(t-1) - C|. \quad (4.19)$$

Equation (4.19) shows that the estimation error of a bad element $|\underline{z}_t - H_t \underline{o}_t|_i$ contains the scene value component of element $i - k$. The scene value $v_{i-k}^c(t)$ changes in time because of the camera’s motion and the change in the value of k . As a result \hat{o}_i can not converge to a specific value (for example C) and the estimation error will remain high. The detection of bad elements is done by averaging over time the estimation error $|\underline{z}_t - H_t \underline{o}_t|_i$. A threshold is then set to differentiate between good and bad elements. This threshold can be set as an absolute value or can be defined statistically from the detector’s behavior. Since element “death” (the process in which an element changes from good to bad) is not frequent the averaging period can be very long. Using a long averaging period reduces the effects of system noise and improves detection reliability. Using this method for bad element detection has the added benefit of detecting “blinking” elements (elements who’s output changes between different constant outputs, usually on and off). This property results from the use of the estimation error and not the element’s absolute output. Figures 11,12 and 13 demonstrate the output after the average in

time of the estimation error of a detector with bad elements. Figures 11 and 12 demonstrate the effects of the various averaging times (100, 150 and 350 iterations) on the estimation error noise. Longer averaging time produces better distinction between good and bad elements. Figure 13 demonstrates the use of a statistical thresholding of 3σ (σ is calculated relative to all elements in the detector) to detect bad elements.

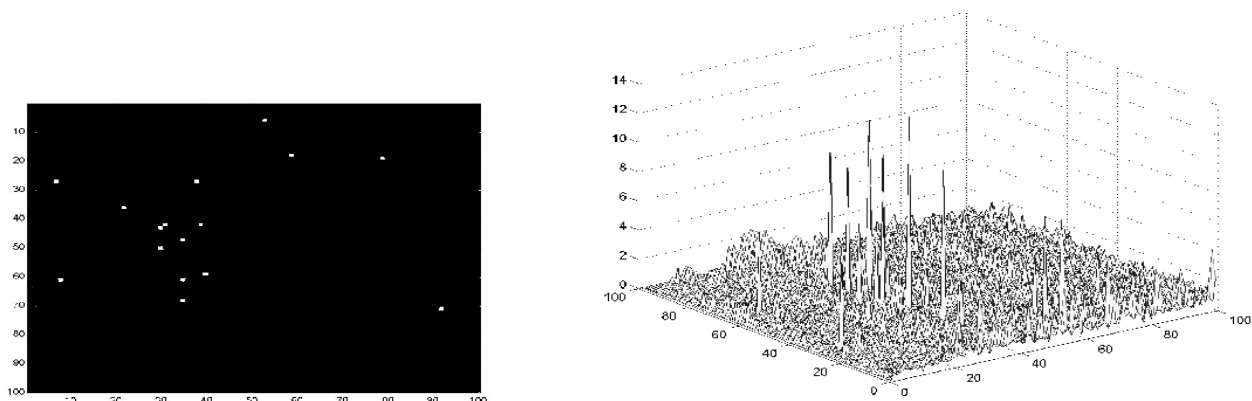


Figure 11: Left picture: Original map of bad elements. Right picture: average value of $|z_t - H_t o_t|$ after 100 iterations.

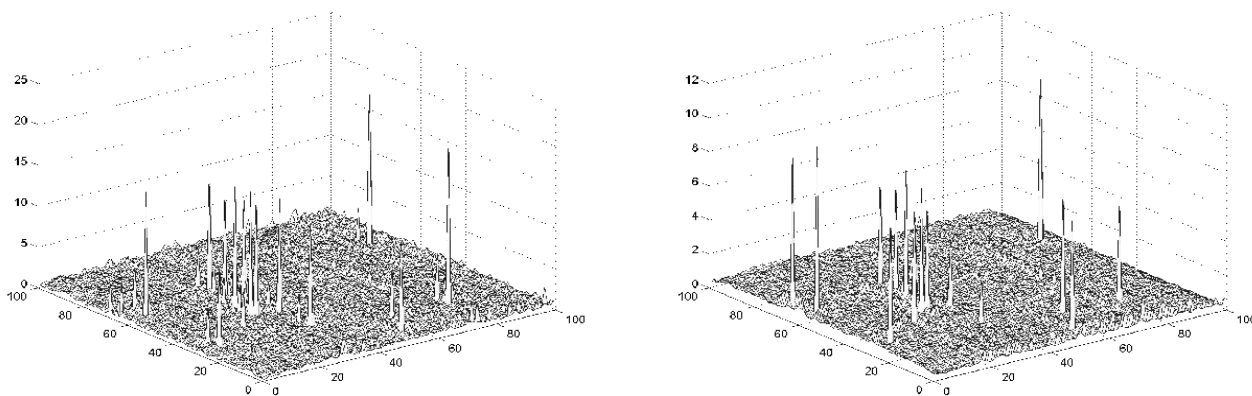


Figure 12: Left picture: average value of $|z_t - H_t o_t|$ after 150 iterations. Right picture: after 350 iterations.

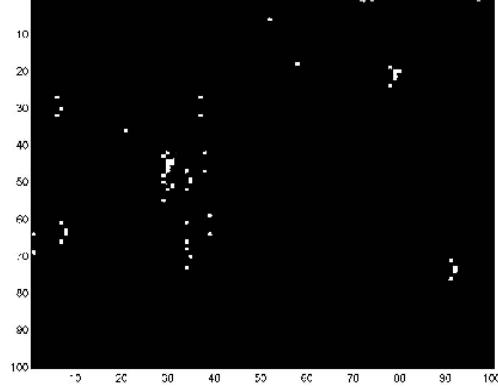


Figure 13: Map of detected bad elements after 150 iterations and a threshold of 3σ .

4.10 Image registration

Video frame registration has attracted a great deal of attention and has been used in variety of applications such as video compression, tracking, computer vision etc. (see for example [20, 21]). The registration model used for defining H_t (see Eqs. (4.15) and (4.17)) and z_t (see Eq. (4.14)) assumes camera motion is made of pan and tilt operation only. This motion model is restrictive, not allowing for camera roll, zoom or non-rigid transformations and was selected for its simplicity.

The proposed correction algorithm is not dependent on the registration model that assumes only pan and tilt movements. The algorithm can support any registration model or technique available with the proper adjustments to the H matrix formulation law. For example, H matrix that supports roll of θ radians (around optical center) is defined by:

assume:

$$\begin{aligned}
 y &= (i \div N) \\
 x &= (i \% N) \\
 r &= \sqrt{(y - \frac{N}{2})^2 + (x - \frac{M}{2})^2} \\
 \alpha &= \tan^{-1} \left(\frac{y - \frac{N}{2}}{x - \frac{M}{2}} \right) + \theta \\
 y' &= \frac{N}{2} + r \cdot \sin(\alpha) - k_y \\
 x' &= \frac{M}{2} + r \cdot \cos(\alpha) - k_x \\
 j' &= y' \cdot N + x'
 \end{aligned}$$

then

$$h(k_y, k_x, \theta)_{i,j} = \begin{cases} 1 & i = j \\ -1 & j = j', \quad 0 \leq y' < N, \quad 0 \leq x' < M \\ 0 & \text{otherwise} \end{cases}$$

Similarly, updates to the corrections method which support affine transformations or even optical flow algorithms can be easily incorporated. It is important to remember that the registration algorithm

needs to function in a noisy environment with correlated noise. The non-uniformity has fixed pattern that tends to affect the registration quality. We suggest that the chosen registration algorithm should either adapt to correlated noise or that appropriate preprocessing is applied on the images.

5 Experimental results

5.1 Development environment

The results which are reported in this section were generated with MATLAB 5.3. The computer used to generate the results was a dual processor Pentium III 550MHz computer, with 256 MB RAM running under WinNT. They implement the correction algorithm which was described in section 4. The sparse data structure of MATLAB was heavily used.

5.2 Source of raw material (input video sequences)

The results are based on video sequence produced by using a 320×240 element thermal camera. The camera's detector is manufactured using *InSb* technology. Detector wavelength is $3 - 5\mu m$. Each pixel is a grey level of 8 bits. Two video sequences were used to test the correction algorithm.

Sequence 1: 1000 frames of urban scenery. The viewing area is approximately 6×6 meters.

Sequence 2: 200 frames of urban scenery. The viewing area is approximately 3×4 meters.

Because the camera that captured the sequences applied two-point non-uniformity correction on the source images, artificial non-uniformity was added to the original sequence in order to simulate a real situation. The artificial non-uniformity added is stationary (does not change in time) and additive (offset only). This type of noise reflects most of the real scenarios. This sequence was then passed to the correction algorithm for removing these added patterns. Three distinct non-uniformity patterns were added as a test-bed of the algorithm.

Pattern 1: Non-uniformity patterns that were captured from the camera.

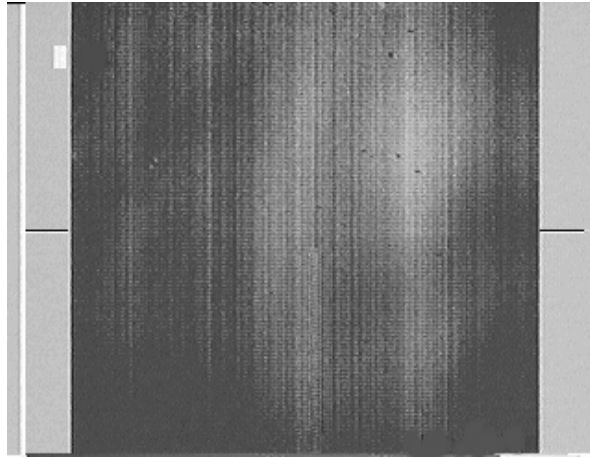


Figure 14: Pattern 1: non-uniformity sampled from thermal camera.

Pattern 2: Synthetic non-uniformity with horizontal features.



Figure 15: Pattern 2: synthetic non-uniformity with strong horizontal features.

Pattern 3: Synthetic pattern that contains features that help to evaluate the spatial frequency.

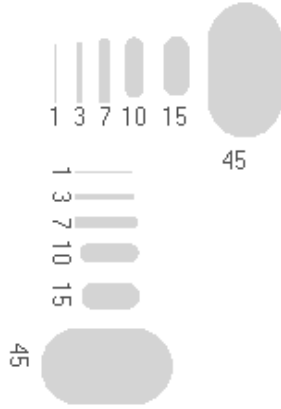


Figure 16: Pattern 3: synthetic pattern used to check correction algorithm's spatial frequency sensitivity.

5.3 Performance evaluation of the correction algorithm

The correction success is measured according to the following criteria: a. Subjective view of the sequence of the corrected frames: We expect that the corrected frames will not contain new artifacts that were not present in the original frame. b. Subjective view of the residual non-uniformity frame: The residual frame contains the difference between the artificially added non-uniformity and the corrected non-uniformity frame. c. Behavior of the of the global residual non-uniformity along time: The global residual non-uniformity is defined as the 2-D standard deviation of the residual frames. Propagation of the residual error, given by standard deviation norm, is visualized in the subsequent graphs. The residual non-uniformity error should decrease with time. It is especially interesting to see the rate of decrease of the residual non-uniformity error. d. Behavior of the of the local residual non-uniformity along time: The local residual non-uniformity is defined as the average of the 2-D standard deviation values calculated on non-overlapping 20 elements windows of the residual frame. Propagation of the local residual error, given by standard deviation norm on small windows, is visualized in the subsequent graphs.

5.3.1 Performance of the non-uniformity correction algorithm

In this section we describe the results of the application of the correction algorithm on two sequences. The correction was applied on a portion of 150×150 elements out of 320×240 elements. The use of a portion of the entire detector is due to computational limitations. A real time variation of this algorithm removes this computational limitation (see section ??).

1. The results in this section are based on processing sequence 1, frames 250 – 450 (out of 1000 frames). Pattern 1 was added to the original sequence. Figures 17 – 25 illustrate the frame

sequence correction. The residual non-uniformity (RNU) of the original non-uniform sequence is 23. After 30 iterations of the algorithm (1 second at 30Hz) the RNU is down to 7.5 (Fig. 25). This is also supported visually by Fig. 22 that has little visible non-uniformity.

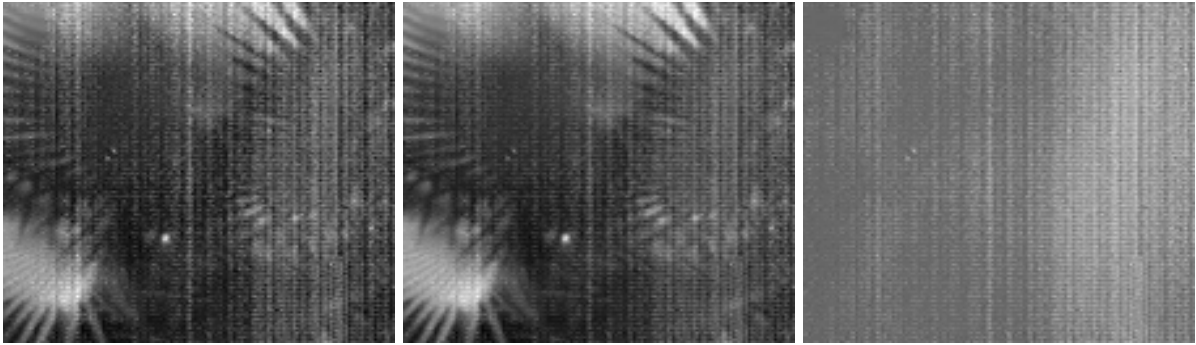


Figure 17: After 1 iteration. Left: non-uniform, center: corrected, right: residual.



Figure 18: After 4 iterations. Left: non-uniform, center: corrected, right: residual.



Figure 19: After 7 iterations. Left: non-uniform, center: corrected, right: residual.

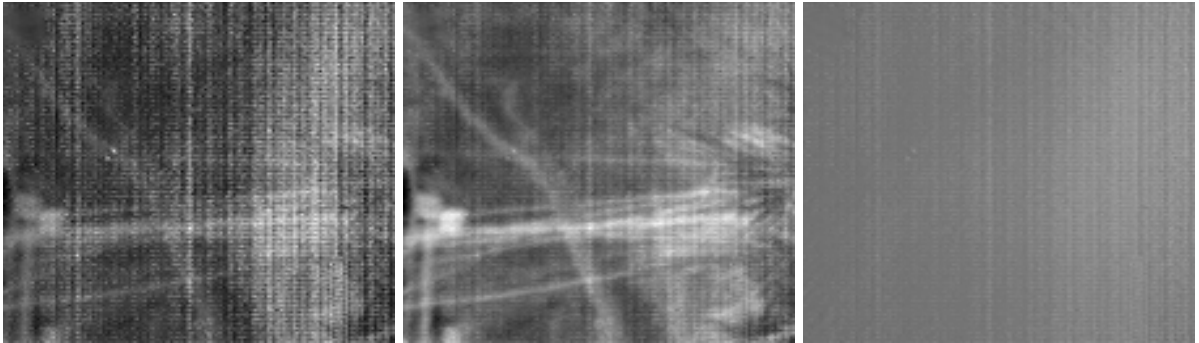


Figure 20: After 10 iterations. Left: non-uniform, center: corrected, right: residual.

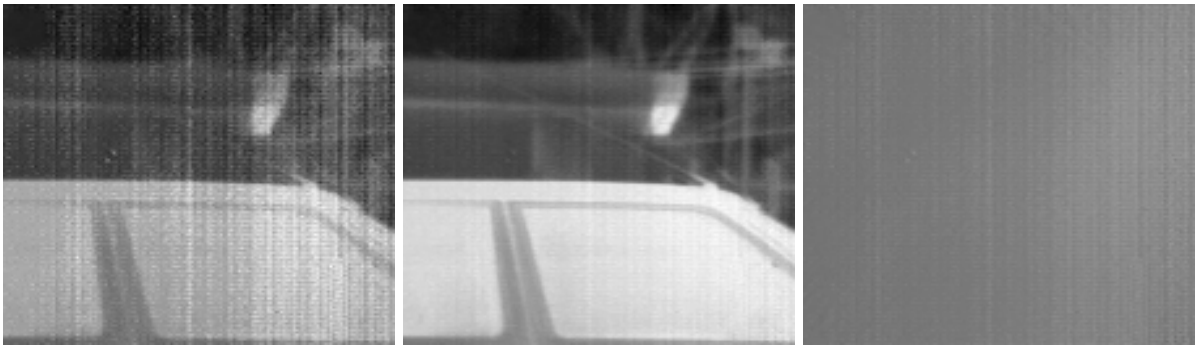


Figure 21: After 19 iterations. Left: non-uniform, center: corrected, right: residual.



Figure 22: After 31 iterations. Left: non-uniform, center: corrected, right: residual.

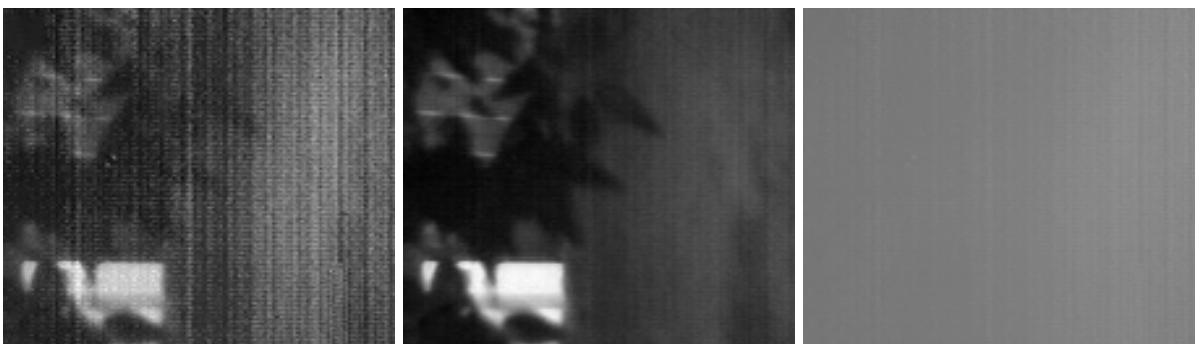


Figure 23: After 61 iterations. Left: non-uniform, center: corrected, right: residual.



Figure 24: After 121 iterations. Left: non-uniform, center: corrected, right: residual.

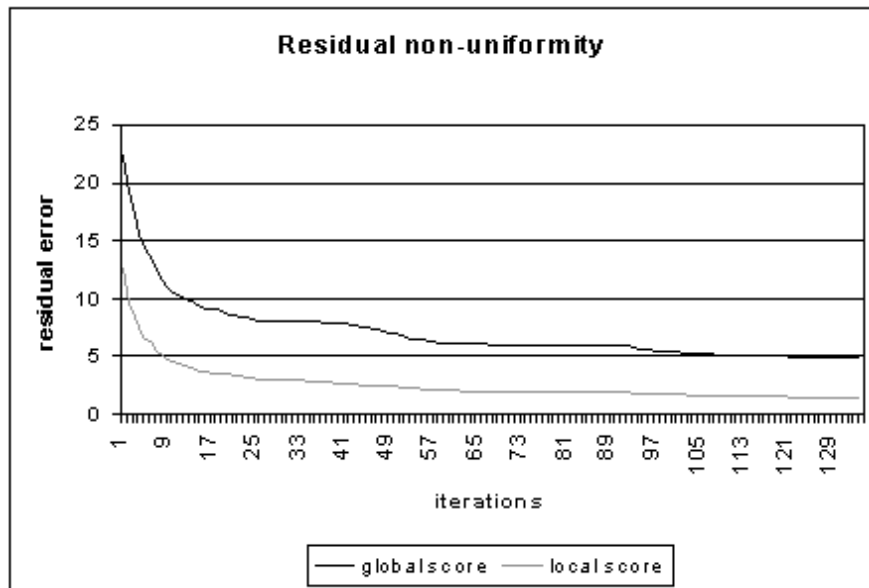


Figure 25: The residual error (standard deviation units) of global and local non-uniformity along time.

2. The results are based on processing of sequence 2 frames 1 – 200 (out of 200 frames). Pattern 2 was added to the original sequence. Figures 26 – 34 illustrate the output of the application of the correction algorithm.

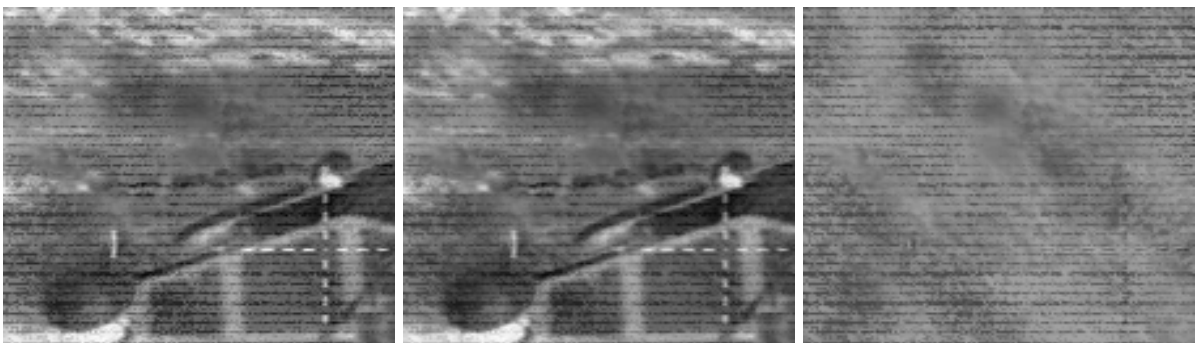


Figure 26: After 1 iteration. Left: non-uniform, center: corrected, right: residual.

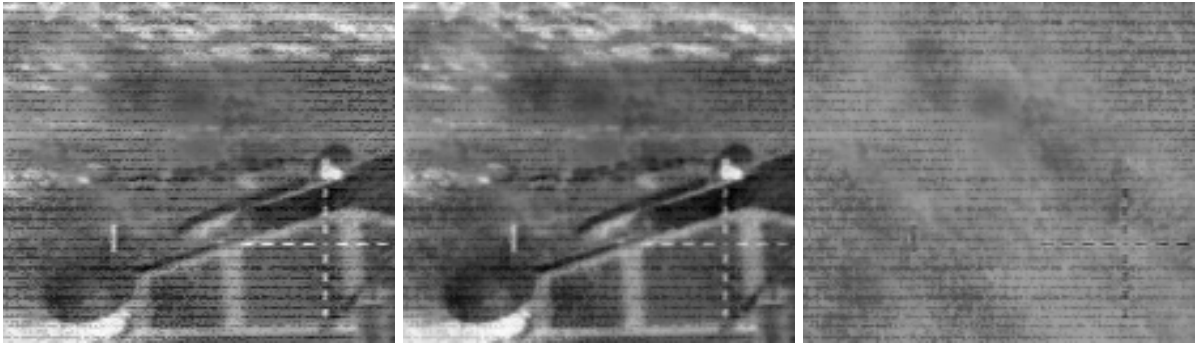


Figure 27: After 4 iterations. Left: non-uniform, center: corrected, right: residual.



Figure 28: After 7 iterations. Left: non-uniform, center: corrected, right: residual.



Figure 29: After 10 iterations. Left: non-uniform, center: corrected, right: residual.



Figure 30: After 16 iterations. Left: non-uniform, center: corrected, right: residual.

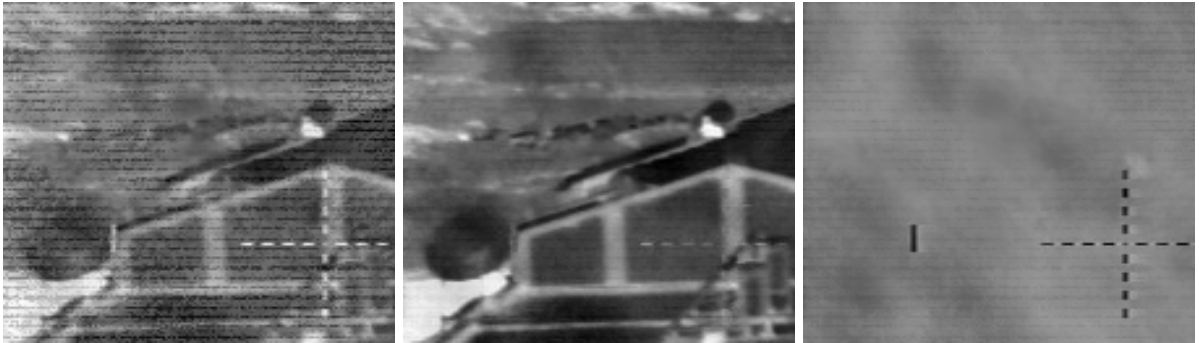


Figure 31: After 31 iterations. Left: non-uniform, center: corrected, right: residual.

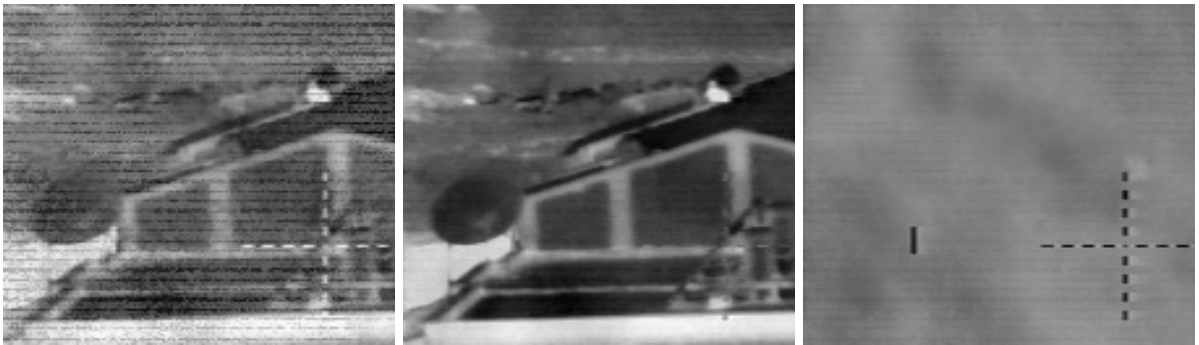


Figure 32: After 61 iterations. Left: non-uniform, center: corrected, right: residual.

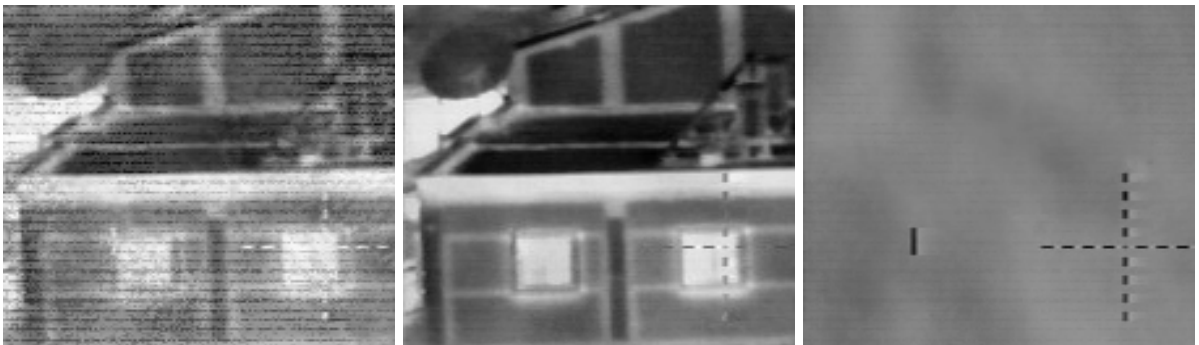


Figure 33: After 121 iterations. Left: non-uniform, center: corrected, right: residual.

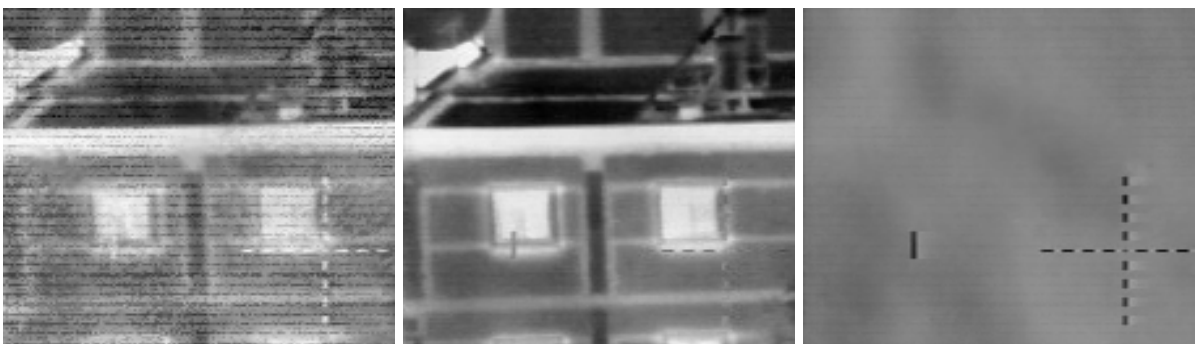


Figure 34: After 166 iterations. Left: non-uniform, center: corrected, right: residual.

3. The results are based on processing sequence 1, frames 250 - 450. Pattern 3 was added to the original sequence. The correction was applied on a portion of 160×160 elements out of 320×240 elements, while using 6×6 tiles with 30×30 elements in each tile. The tile overlap width was set to 4 elements. As can be seen in Figs. 35 - 37 the non-uniformity of all spatial frequencies was corrected.



Figure 35: After 7 iterations. Left: non-uniform, center: corrected, right: residual.



Figure 36: After 31 iterations. Left: non-uniform, center: corrected, right: residual.

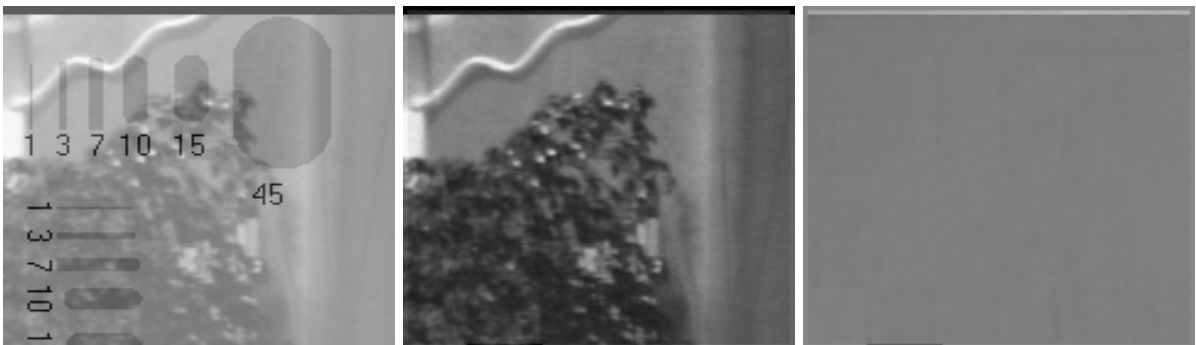


Figure 37: After 131 iterations. Left: non-uniform, center: corrected, right: residual.

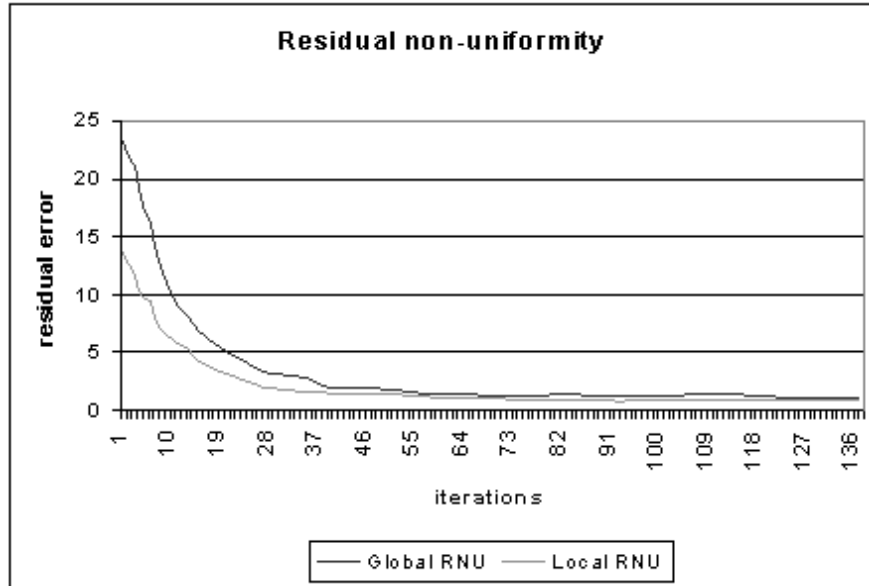


Figure 38: Residual error (standard deviation units) of global and local non-uniformity along time. We can see that as the number of iterations increases the residual error decreases.

6 Conclusions and future work

This paper introduces an algorithm for correcting one point non-uniformity. The correction algorithm is scene based and relies only on the camera’s video sequence for the correction procedure. The correction is accomplished by utilizing image registration and Kalman filtering. The Kalman filter is used in order to track the correction values evolved in time. The algorithm does not utilize any low pass filters that reduce the performance of the detector. No visible artifacts or unwanted features are produced after the application of the algorithm. The correction algorithm has a high convergence rate. Experimental results indicate that most of the non-uniformity is removed within 30 iterations of the algorithm (less than 1 second at sampling rate of $30Hz$). A window based variant of the baseline Kalman correction algorithm was also introduced. This modification improved the algorithms execution speed without demaging the correction quality. A parallel implementation of this window based correction algorithm is also possible. Both the base-line and the window based correction algorithms have a linear memory and computational requirements.

This work can be extended to a number of directions. The correction algorithm and the correction program can be extended to support more complex registration models like affine transformations and optical flow. It is also interesting to extend the algorithm to support non-stationary scenes (scenes that contain moving objects). The non-uniformity correction model can be extended from offset correction to linear correction and even to non-linear models. This direction is relevant for the implementation of a Kalman based correction to different types of cameras and to different types of distortions. Another

possible research direction is the development of a realtime implementation of this algorithm (probably hardware based).

References

- [1] Schultz M., Caldwell L., “Nonuniformity correction and correctability of infrared focal plane arrays, Proc. SPIE vol. 2470, pp. 200-211, April 1995.
- [2] Dierickx B., Meynants G., “Missing pixel correction algorithm for image sensors, Proc. SPIE vol. 3410, pp. 200-203, May 1998.
- [3] Gross W., Hierl Th., Schultz M., Haigh J., “Correctability of the spatial nonuniformity in various infrared focal plane arrays, Proc. SPIE vol. 3436, pp. 203-213, July 1998.
- [4] Scribner D. A., Sarkady K., Kruer M.R., Caulfield J.T., Katz G., Gridley, C.J. “Nonuniformity Correction for Staring IR Focal Plane Arrays Using Scene Based Techniques”, Proc. SPIE Vol. 1308, April 1990.
- [5] Scribner D. A., Sarkady K., Kruer M.R., Caulfield J.T. “Adaptive Nonuniformity Correction for IR Focal Plane Arrays Using Neural Networks”, SPIE Vol. 1541, 1991.
- [6] Hefpner K., Springs C., Horman R. “Method and device for improvement with compensation for individual detector response, U.S. Patent no. 5276319, 1994.
- [7] Venkateswarlu R., Er M. H., Gan Y. H., Fong Y. C., “Nonuniformity compensation for IR focal plane array sensors”, Proc. SPIE vol. 3061, pp. 915-926, 1997.
- [8] Harris J.G., Chiang Y.M., “Nonuniformity correction using constant-statistics constraint: analogue and digital implementations”, SPIE Vol. 3061, pp.895-905, 1997.
- [9] Chiang Y.M., Harris G.H., “An analog integrated circuit for continuous time gain and offset calibration of sensor arrays”, J. of analog integrated circuits and signal processing, Vol. 12, pp. 231-238, 1997.
- [10] Harris J.G., Chiang Y.M., “Minimizing the ghosting artifact in scene based nonuniformity correction”, SPIE Vol. 3377, pp.106-113, 1998.
- [11] Sophia T., Alan H. L., “Scene based techniques for nonuniformity correction of infrared focal plane arrays”, SPIE Vol. 3436, Infrared Technology and Applications XXIV, pp.172-183, 1998.
- [12] ONeil W. F., “Dither image scanner with compensation for individual detector response and gain correction”, U.S Patent No. 5514865, 1996.

- [13] Irani M., Peleg S., “Improving resolution by image registration”, *CPIG: Graphical Models and Image Processing*, vol. 57, no. 3, pp. 231-239, 1991.
- [14] Irani M., Peleg S., “Image sequence enhancement using multiple motions analysis”, *IEEE conference on computer vision and pattern recognition*, June 1999.
- [15] Irani M., Peleg S., “Motion Analysis for Image Enhancement: Resolution, Occlusion, and Transparency”, *J. Vision Communication and Image Rep.*, vol. 4, no. 4, pp. 324-335, Dec. 1993.
- [16] Golub G. H., Van Loan C. F., “*Matrix Computations*”, Jhon Hopkins University Press, 1983.
- [17] Elad M., Feuer A, “Restoration of a single super-resolution image from several blurred, noisy and under sampled measured images”, *IEEE transactions on image processing*, Vol. 6, pp. 1646-1658, Dec. 1997.
- [18] Elad M., Feuer A, “Super-resolution reconstruction of image sequences”, *IEEE transactions on pattern analysis and machine intelligence*, Vol. 21 No. 9, pp. 817-834, sept. 1999.
- [19] Gelb Arthur, “*Applied Optimal Estimation*”, M.I.T press,1999.
- [20] De Castro E. and Morandi C. , “Registration of translated and rotated images using finite Fourier transform”, *IEEE Trans. Pattern Anal. Machine Intell.*, PAMI-9(5):700-703, 1987.
- [21] Reddy B. S. and Chatterji B.N., “An FFT-based technique for translation, rotation and scale-invariant image registration”, *IEEE Trans. Image Proc.*, 5(8):1266-1271, 1996.