Real-Time Implementation Issues in Passive Automatic Focusing for Digital Still Cameras

V. Peddigari, M. Gamadia and N. Kehtarnavaz†
Department of Electrical Engineering, University of Texas at Dallas, Richardson, Texas, USA

Due to consumers' demand for faster picture shot time in the rapidly expanding digital still camera market, it is of importance to address the real-time implementation issues in the development of passive automatic focusing for digital still cameras. This article discusses such real-time implementation issues that are often overlooked when designing passive contrast sensing automatic focusing on digital still camera processors. Specifically, algorithmic design tradeoffs between automatic focusing speed, accuracy, and power consumption, are addressed. A sample implementation and its performance results on an actual digital still camera hardware platform powered by the Texas Instruments TMS320DM270 processor are presented to further convey these real-time implementation issues.

Journal of Imaging Science and Technology 49: 114–123 (2005)

Introduction

The market for digital still cameras (DSCs) has experienced a considerable growth in recent years, with sales already passing those of traditional film cameras in 2003 and projected sales of over fifty-one million units by 2007.[1] This rapid growth is driven primarily by advancements in DSC technology coupled with consumers' desire to view and transfer images instantaneously.

Unlike traditional film-based cameras, digital cameras produce a full color image from captured raw CCD/CMOS sensor data via a number of image processing modules, collectively referred to as the image pipeline.[2] Figure 1 illustrates a typical image pipeline of a digital still camera. As illustrated in Fig. 1, in the pre-capture processing stage, the CCD/CMOS sensor data is continuously read and digitized while altering exposure, focus, and white balancing.[3] Auto-Exposure (AE), Auto-Focus (AF), and Auto-White-Balance (AWB) are the algorithms used during this stage. After the correct focus has been obtained by the AF algorithm, commonly referred to as AF lag time, a picture is captured. As indicated in Fig. 1, in the post-capture processing stage, appropriate image processing steps are then taken to enhance the captured raw CCD/CMOS sensor data, producing a full color image which is finally compressed and stored onto flash memory.

There are two main approaches to automatic focusing: active AF and passive AF. Active AF makes use of infrared or ultrasound distance measuring sensors to adjust the focus lens and bring the image into focus. Passive AF, on the other hand, does not utilize any distance measuring sensor, but rather extracts sharpness information from the image itself to adjust the focus lens and bring it into focus. So called contrast sensing is a popular passive AF method that is most

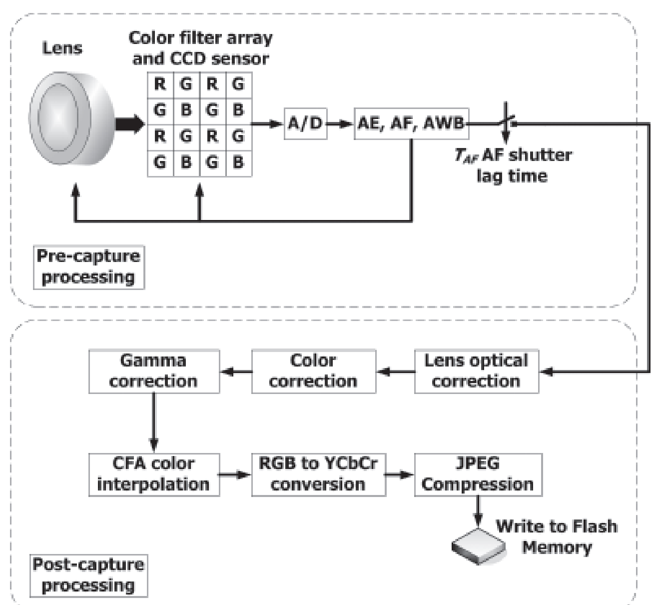

Figure 1. Typical digital still camera image processing pipeline.

Original manuscript received July 14, 2004

†Corresponding Author: N. Kehtarnavaz, kehtar@utdallas.edu

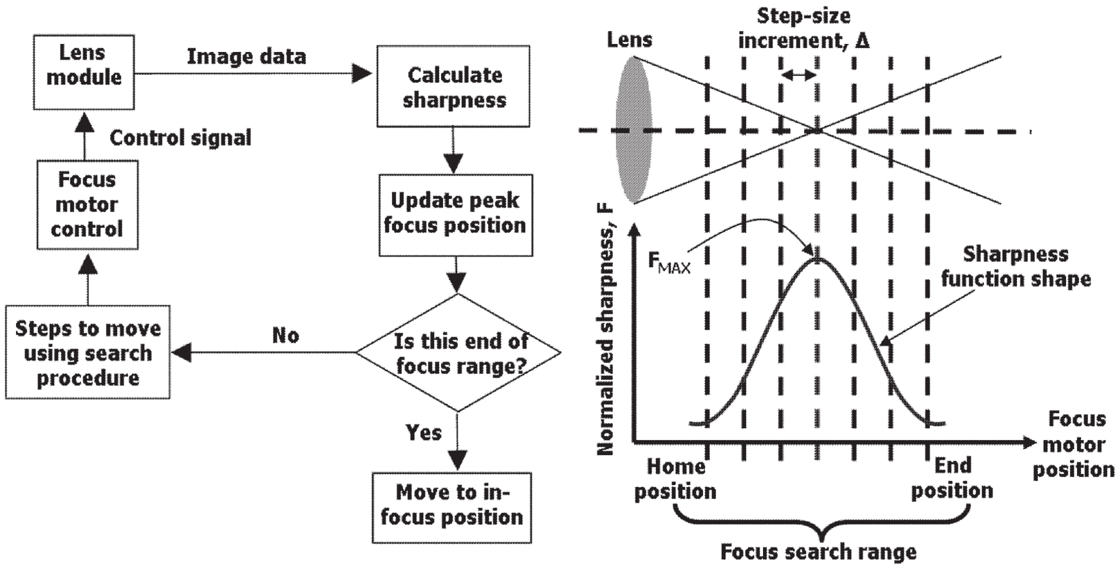

Figure 2. Flowchart of generic contrast sensing passive AF algorithm.

widely used in consumer digital cameras, primarily due to its cost effectiveness and software flexibility.

Many passive contrast sensing AF algorithms have been introduced in the literature.[4–10] However, one aspect of these AF algorithms that has not been adequately addressed or missing in the literature is the real-time implementation issues as related to performance tradeoffs between focusing speed, accuracy, and power consumption. The work presented in this article is an attempt towards addressing such real-time implementation issues that one needs to be aware of when designing passive AF systems.

This article is organized as follows. The next section provides an overview of the three essential components common to all contrast sensing passive AF algorithms. We then discuss the design choices behind each of the three components exhibiting how such choices affect real-time AF performance. Real-time implementation of our passive AF algorithm on the Texas Instruments TMS320DM270 (DM270) digital camera processor is presented along with obtained performance results, and finally, the conclusions are stated.

How Contrast Sensing Passive AF Works

There are three essential components to any passive AF algorithm. These components consist of focusing region or regions, measure of sharpness, and peak search procedure.

In a passive AF system, first it is required to choose a focusing region, from which the image sharpness information is computed. Second, a sharpness function is utilized to provide the degree of image focus or sharpness in the focusing region. Finally, a peak search procedure is applied to obtain the highest sharpness value while varying the focus lens. The entire passive AF process is done in an iterative manner, where the focus lens is controlled by a stepper motor. Figure 2 shows a flowchart of the generic iterative passive AF algorithm.

Initially, the stepper motor begins the search at a starting lens position and a sharpness value is computed for this position. The lens is then moved to a next position, where another sharpness value is computed. This process is continued until the position yielding the maximum sharpness value is identified. This position corresponds to the in-focus lens or focus motor position. It should be noted that a next position is decided based upon the step size increment provided by the search procedure. The movement of the stepper motor is achieved by a lens control module.

Real-Time Implementation Issues

Although the components of passive AF algorithms are easy to comprehend, their implementations on a digital camera hardware platform are not so straightforward. Depending on the type of hardware platform available, there are many implementation issues which have to be taken into consideration in order to achieve a balance between focusing speed, accuracy, and power consumption. The major implementation issues of concern are: What focusing region size and how many focusing regions to use? Which sharpness function to consider? What search procedure to adopt? These issues are discussed in the subsections that follow.

Focusing Region

Considering that passive AF systems extract sharpness information from the image data itself, one needs to decide what portion of the image to use for this computation. A digital camera processor often imposes computational limitations which affect the speed of the AF algorithm.

A widely used focusing region consists of a center area within the image frame. This region should be large enough so that the object of interest fully or partially falls in it. As the size of the focusing region is made larger, the computational time for computing a sharpness function increases. If the digital camera processor has a dedicated component for calculation of the sharpness function, any focusing region size can be considered.

In practice, since users normally tend to frame their objects of interest within the center of the image, the focusing region is limited to only a small central area.[11] This way the computation time is considerably reduced. As depicted in Fig. 3, such focusing regions are often used in video camcorders, and in some digital cameras, where a larger focusing region is used to surround a smaller focusing region at the center of the image.[5–7]

Figure 3. Center focusing regions commonly utilized in contrast sensing passive AF algorithms.

A center focusing region can be effective for most photographic compositions, but it does not always work when the object of interest is placed outside the center area, for example. As a result, the camera ends up focusing on the background, leaving the object of interest out-of-focus. This problem can be resolved by increasing the size of the focusing region. Some camera manufacturers provide options for users to place the focusing region almost anywhere within the image frame to resolve this problem manually.[12]

It should be realized that as the focusing region becomes larger, multiple objects, located at varying distances from the camera, may contribute to the sharpness function. Depending on the depth of field, this can produce multiple peaks in the sharpness function, each peak corresponding to a specific object at a specific distance or plane of focus. Multiple peaks can cause the search procedure to choose a wrong plane of focus. Thus, in order to isolate peaks corresponding to different planes of focus, a single focusing region is often divided into multiple focusing regions, and a decision algorithm and/or weighting scheme is used to decide on which plane to focus.

Sharpness Function

Passive AF systems use a sharpness function to calculate the degree of focus from a focusing region. Various sharpness functions have been introduced in the literature.[4–10,13–19] In general, sharpness functions reflect the degree of high-frequency or edge detail within a focusing region. The most popular sharpness functions are designed using gradient operators, which are inherently high-pass filters commonly used for edge detection. Due to the possibility of getting multiple peaks, an appropriate criterion for selecting a sharpness function is that it should produce a well defined peak corresponding to the in-focus position of the object of interest.

In practice, only the green portion of the sensor data, which effectively represents the image luminance, is often used in order to save the computation time associated with processing all the three primary colors.[9,10] The use of green component requires carrying out a proper indexing corresponding to the color filter array that is placed in front of the CCD sensor.

Among the gradient based sharpness functions, the squared gradient measure has been shown to provide well-defined peaks.[10,17] This measure, denoted by F_s, simply sums the squared differences between adjacent pixels,

$$F_s = \sum_{i=1}^{M} \sum_{j=2}^{N} \left[y_s(i,j) - y_s(i,j-1) \right]^2 \tag{1}$$

where $y_s(i,j)$ indicates the luminance of a pixel in the ith row and jth column of a focusing region, consisting of M rows and N columns, taken at a focus motor position s. The squaring allows more weight to be applied to larger gradients.

Another widely used sharpness measure is one-dimensional digital filtering of the rows within a focusing region.[6,9] This measure provides more selectivity since the filter coefficients can be specified to produce various types of filter responses such as high-pass or band-pass. This measure is sensitive to vertical edge detail, since the filtering is performed along horizontal scan lines. It is also possible to apply the same filtering process vertically to enable detection of horizontal edge detail.

In general, sharpness functions can be implemented in software and executed on the digital camera processor. However, depending on the size of the focusing region or regions and the speed of the system memory, such an implementation can be quite slow. Gain in focusing speed can be achieved by implementing the sharpness function in a dedicated hardware coprocessor rather than in software. When it comes to decide what type of sharpness measure to adopt, the preliminary testing can be performed in software. Once an effective measure is decided upon, it can then be implemented in hardware to generate a lower AF lag time.

Of course, it should be noted that passive AF systems cannot focus under inadequate lighting conditions or when the object of interest lacks sufficient contrast or edge detail. Such situations result in a relatively flat sharpness function, which should not be used in determining the in-focus position. A simple way to detect a flat sharpness function is by examining the percentage difference between the minimum and maximum values, which stays quite small for a flat sharpness function. Thus, upon detection of a flat sharpness function, the peak position should be ignored and the focus lens be moved to a pre-determined position, while giving a warning to the user that the scene lacks sufficient contrast for focusing purposes.

Search Procedure

A passive AF algorithm is expected to determine quickly the in-focus position by using either one sharpness function if one focusing region is used, or several sharpness functions, if multiple focusing regions are used. Therefore, it is important to utilize a peak search procedure whose search parameters can be adjusted to achieve a balance between focusing speed, accuracy, and power consumption.

The standard search procedure consists of a full sequential scan through the entire focus range of a given zoom depth from a start position to an end position. This search procedure is known as the global search (GS). Clearly the GS has an advantage in focusing accuracy due to the fact that it is essentially an exhaustive scan across an entire focus range. Thus, the GS can be used to judge the accuracy of other search procedures.

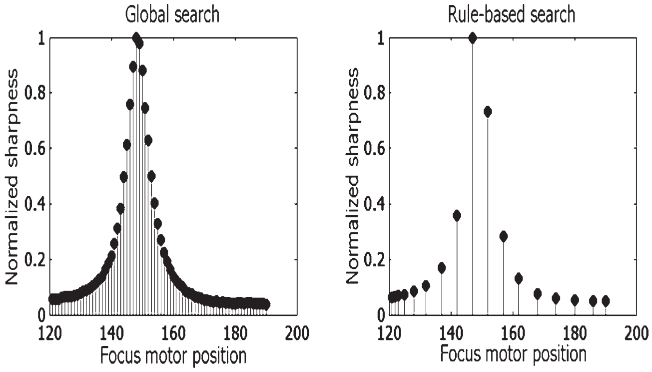

Figure 4. Step size differences between global and rule-based search procedures.

Another advantage of the GS is that it can support multimodal sharpness functions. If a sharpness function has multiple peaks, a decision can be made to determine which peak corresponds to the in-focus position. Some common peak decisions include focusing on the nearest object, the farthest object, or somewhere in between. It should be noted that while the GS is generally slow due to its full scan nature, it can be an effective approach if the amount of steps within the search range is fairly small. In case of larger focus ranges, the use of the GS becomes prohibitive as moving the motor through a full scan significantly increases AF lag time.

One of the critical real-time issues here is the reduction of AF lag time while maintaining a high level of accuracy and low amount of power consumption. Therefore, there have been many attempts to improve upon the standard GS by reducing the number of movements of the stepper motor.[4,5,7–10] Such approaches include divide and conquer algorithms such as the Fibonacci search and the Binary search. The divide and conquer approaches assume a unimodal sharpness function and are based on frequently changing the direction of the stepper motor while reducing the search range in order to reach the peak at a faster rate. While these approaches are computationally efficient, if the stepper motor used requires a gear backlash compensation for changing direction, AF lag time is considerably increased. Hence, when using stepper motors with gear backlash, it is necessary to scan the focus range in a sequential manner in one direction and limit the amount of direction changes in order to reduce the contribution of the stepper motor to AF lag time and system power consumption.

To account for a stepper motor with gear backlash, in Kehtarnavaz[10] a sequential search algorithm was introduced, named the rule-based search (RS). The RS was developed to lower AF lag time while supporting multimodal sharpness functions. As illustrated in Fig. 4, the RS quickly scans through the entire focus range by changing the step size increment according to the rate of change of the sharpness function. Three fixed step size increments are used: Fine, Mid, and Coarse. Mid and Coarse step sizes are used to quickly pass through the portion of the focus search range before and after an in-focus position, while the Fine step size is used when the motor is within the vicinity of an in-focus position. The results reported in Kehtarnavaz[10] showed that the RS improved AF as compared to the GS and the Binary search in terms of AF lag time and power consumption.

Of course, by not searching through every step motor position, it is possible that the in-focus position is offset from the true focus position. As a result, the accuracy of the search procedure suffers when attempting to reduce AF lag time. Adjustment of step sizes or interpolation techniques can be used to further improve the accuracy while reducing AF lag time.

Another real-time issue of concern is the focus range over which the search is performed. That is each zoom depth has its own unique range of positions in which the in-focus position should be found. Hence, for a given zoom depth, it is necessary to search through only a portion of the entire focus range. Also, depending on which direction the stepper motor is designed to move without gear backlash compensation, the focus range can be scanned from a far focus position to a near focus position, or vice versa.

Since the majority of scenes are composed of objects not so close to the camera, it is unnecessary to search through the focus motor positions corresponding to close distances to the camera. Thus, it is more efficient to split the focus search range for each zoom depth into a “Normal” and a “Macro” photographing mode. By allowing the user to select between a Normal versus a Macro mode, appropriate focus search ranges can be selected leading to a significant reduction in AF lag time for both far and close up shots.

Now tying together all the processes involved in the search procedure requires performing proper event synchronization. Hence, one of the main real-time constraints in AF is the synchronization of the events involved in the AF algorithm such as movement of the focus motor, exposure of the image sensor, and calculation of the sharpness function. The sequences of the events are fixed in the order just mentioned, but a proper timing synchronization among the events should be put in place in order to ensure correct AF operation. In the following section, the AF implementation on an actual digital camera processor is presented, which takes into consideration the real-time issues discussed in this section.

Passive AF Implementation on DSC Hardware Platform

This section presents the implementation of our developed passive AF algorithm using the Texas Instruments DM270 processor to convey the real-time issues within the framework of an actual digital still camera hardware platform. In addition to the above discussed real-time issues, other real-time constraints more specific to this hardware platform are mentioned. The performance results corresponding to our extensive indoor and outdoor testing of the developed AF algorithm are also presented.

Real-Time Hardware Platform Constraints

The hardware platform, including the image sensor, the lens module, and the processor, govern the design of a passive AF algorithm. Given the DM270 processor coupled with a standard lens module as the hardware platform, the first step in designing a passive AF algorithm is to assess their respective hardware features.

The DM270 processor, whose architecture is illustrated in Fig. 5, has an AF engine or coprocessor which allows the real-time computation of those sharpness functions which can be implemented via digital filters.[20] This engine

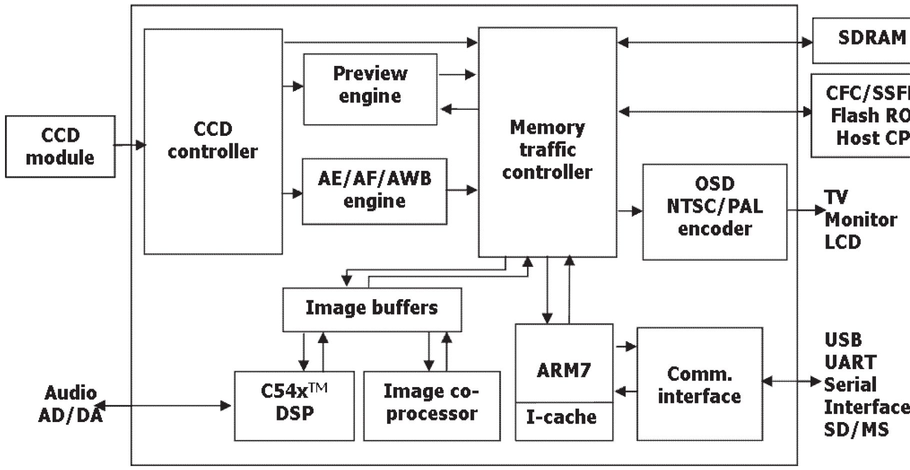

Figure 5. Block diagram of TMS320DM270 processor architecture.[19]

utilizes programmable coefficient IIR digital filters to calculate an AF sharpness function from the green pixels within a focusing region. The focusing region can be programmed, meaning that the sharpness function can be extracted from almost anywhere within the image frame. However, as a result, there are restrictions placed on the sharpness function and focusing region. The sharpness function is limited to the green pixel data and to the order and type of the programmable filters. The location, size and shape of the focusing region are governed by the AF engine. This processor also provides two SDRAM buffers for a quick access of sharpness function data from two consecutive frames.

The standard lens module employed also creates constraints on the AF algorithm. This module includes a stepper motor for the adjustment of focus by varying the distance between the lens and the image sensor over 371 possible step positions. Step position 0 corresponds to focusing of far objects, while step position 370 corresponds to focusing of near objects. The lens module also includes a stepper motor for adjustment of the zoom depth. Depending on the number of zoom depths to be used in a camera, a reduced focus search range for each zoom depth can be experimentally found to further decrease AF lag time. The lens module utilized is equipped with a 3.3 megapixel Sony CCD image sensor which provides two high frame rate readouts, one at 30 frames per second with 258 output lines, and one at 60 frames per second with 96 output lines taken from the central portion of the image. Note that the choice of the frame readout rate restricts the size of the focusing region used.

Focusing Region: "Multiple-Window" Approach

When it comes to designing a focusing region, every choice has its advantages and disadvantages. The center only approach worked well for most photography situations, but it could not satisfy all situations such as when the object of interest was placed off-center. To accommodate for such situations, a wider focusing region was used. Consequently, this required the image sensor frame rate to be set to 30 frames per second for the wider focusing region.

When using a wider focusing region, the possibility of getting multiple peaks increases. To provide a mechanism for isolating the in-focus peaks corresponding to different planes of focus, the focusing region was split into three separate focusing windows: Left window, Center window, and Right window, as illustrated in Fig. 6(a). This demanded putting in place a more involved search procedure to select one of these windows for focusing.

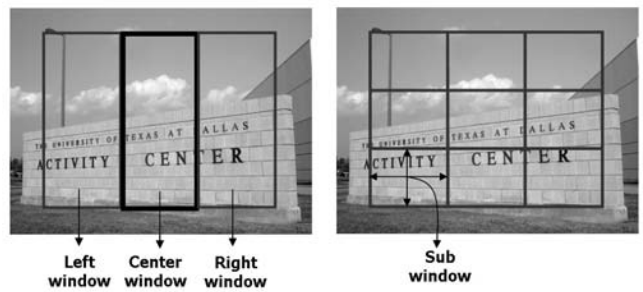

Figure 6. Multiple-window focusing regions to cope with multiple objects: (a) three windows; and (b) nine sub-windows.

Initially, another real-time implementation issue needed to be resolved. Due to the size of the windows, the sharpness functions exhibited the blending of the in-focus peaks resulting in a single in-focus position. This was most prevalent for scenes with a low contrast foreground object over a high contrast background. As a result, it became difficult to distinguish between the in-focus position corresponding to the background versus that of the object of interest in the foreground. Hence, another round of splitting was performed in order to better isolate the in-focus positions. This time, each window was divided into three separate focusing regions called sub-windows, for a total of nine sub-windows, as shown in Fig. 6(b). Here we refer to this approach as the multiple-window approach. It should be noted that the number of sub-windows was decided based on the observation that smaller focusing regions generated noisy sharpness functions, and larger focusing regions led to blending of in-focus positions for multiple objects. For simplicity of implementation, nine sub-windows were thus chosen to avoid the above mentioned drawbacks.

By splitting a window into three sub-windows, it became possible to isolate the in-focus position of the object of interest. For example, Fig. 7 depicts the

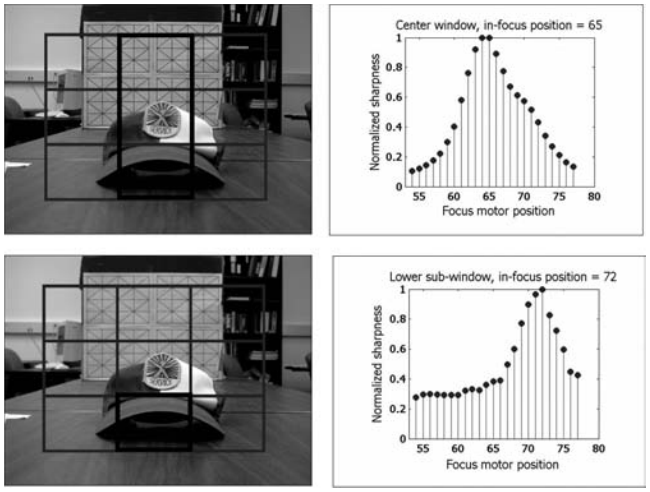

Figure 7. Blending of in-focus positions when using a wider focusing region.

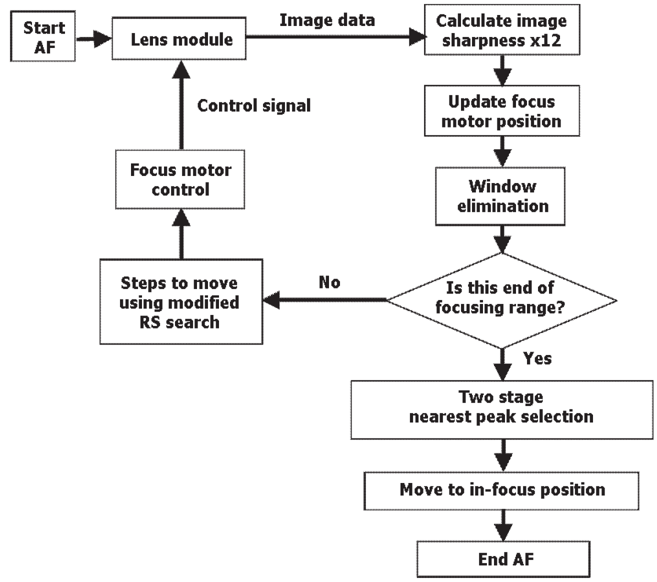

Figure 8. Flowchart of the developed multiple-window contrast sensing passive AF algorithm.

resulted blending in the sharpness function of the Center window as compared to that of its lower sub-window. The Center window shows an in-focus position of 65 for the high contrast background while the sharpness function of its lower sub-window shows an in-focus position of 72 for the relatively lower contrast object of interest, the hat. By using the multiple-window approach, the focusing accuracy was improved since it allowed better isolations of the in-focus positions.

It should be realized that the multiple-window approach increases both the complexity and computational cost of any search procedure, since it is now required to compute and compare the sharpness functions for the three windows and nine sub-windows.

Sharpness Function: DM270 AF Filter

Although it was possible to implement the squared-gradient sharpness function in software, it was found that the gain in the computational speed of the DM270 AF engine outweighed that of the software implementation. The use of the AF engine required determining appropriate filter coefficients. Based on our study reported in Gamadia,[18] it was obtained that a high-pass filter, named the first difference filter, provided the highest focusing accuracy among many filters that were examined. The impulse response of this filter is given by:

$$h(j)=\begin{cases}1, j==0\\-1, j=1\\0, elsewhere\end{cases} \xleftrightarrow{Z-transform} H(z)=1-z^{-1} \tag{2}$$

The sharpness function is calculated using the following expression,

$$F_s=\left(\sum_{i=1}^{M}\max_{j\in[1,N]}\left|y_s(i,j)*h(j)\right|\right)^2 \tag{3}$$

where * denotes the discrete convolution operator and $y_s(i,j)$ the luminance intensity of a pixel in the ith row and jth column of a focusing region consisting of M rows and N columns obtained at a step motor position s. The max operation and squaring are done to emphasize larger gradients over smaller ones, similar to the squared-gradient sharpness function.

Search Procedure: Modified Rule-Based Search

The search procedure has a major impact on the overall speed and accuracy of a passive AF algorithm. Due to the use of multiple focusing regions and sharpness functions, the AF algorithm flowchart depicted in Fig. 2 was modified to accommodate the multiple-window approach.

Developed AF Algorithm Overview

Noting that the lens module employed required the use of gear backlash for movement of the stepper motor from a near position to a far position, it was decided to perform the search starting from a far position to a near position for a given zoom depth in order to reduce the effect of the stepper motor movement on the AF lag time. In addition to this, because of the gear backlash, the use of divide and conquer search procedures were ruled out for the same reason. Even if there was no required gear backlash compensation, it would not make sense to use such search procedures since they only support unimodal sharpness functions, which do not always occur. Because of the adjustable parameters of the sequential rule-based search procedure and the fact that it can support multimodal sharpness functions, this procedure was used as our search procedure and modifications were made to it in order to support the use of multiple focusing regions and to further reduce the AF lag time. Figure 8 shows the flowchart of our developed two-stage passive AF algorithm using the modified rule search procedure.

The rule-based search controls step size increments according to the rate of change of a single sharpness function. Considering that there were multiple sharpness functions, an assumption was made to use the sharpness function of the object closest to the

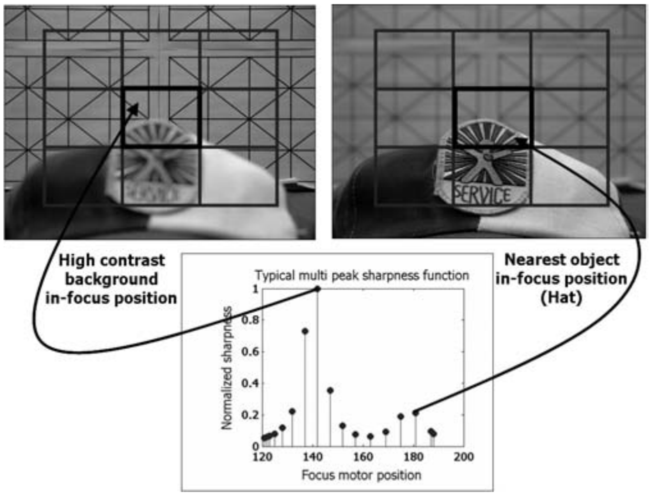

Figure 9. In-focus position for the maximum sharpness value (background) versus that for the nearest object (hat).

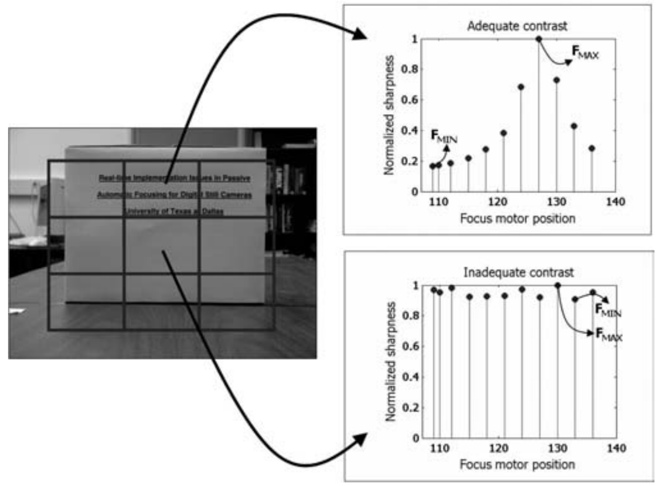

Figure 10. Sharpness functions in the presence of adequate and inadequate contrast.

camera. This assumption is justified because in most cases camera operators intend to focus on objects which are closer to the camera rather than farther from the camera. It is worth mentioning that, if desired, our setup allows one to weigh the sharpness functions differently. Figure 9 illustrates the advantage of using the in-focus position of the nearest object (hat) as compared to the conventional approach of simply using the maximum as the in-focus position (background).

With this in mind, the RS step size increment decision was done based on the window(s) having an in-focus position for the nearest object. This is shown as the Window Elimination module in Fig. 8. As the search procedure scans through a focus search range, the Window Elimination module keeps track of the rise and fall of the sharpness functions of the three windows for the purpose of setting a flag. If one of the three windows exhibits an in-focus position corresponding to a farther object, the flag is set for that window. As a result, its sharpness function is ignored in deciding the step size. In our case, the Left, Center and Right windows were used for window elimination purposes, as the use of the nine sub-windows for window elimination would have required 2^9 comparisons as compared to 2^3 comparisons for the three windows.

After the stepper motor reaches the end of the focus search range, the second stage of the AF algorithm begins. First, the window having the in-focus position corresponding to the nearest object is selected. Then, in order to pick out a correct in-focus position corresponding to the nearest object, even in cases of high contrast backgrounds and low contrast objects, the sharpness functions of the three sub-windows within the selected window are compared to identify the nearest in-focus position.

If the object of interest lacks sufficient contrast, certain sharpness functions might exhibit a flat behavior and thus are not effective for determining the in-focus position. To make the in-focus determination more robust, another flag was used to indicate if the sharpness function of a window or sub-window was flat based on the percentage difference between their respective maximum and minimum sharpness samples. Figure 10 shows an example of the sharpness function for a low and a high contrast area.

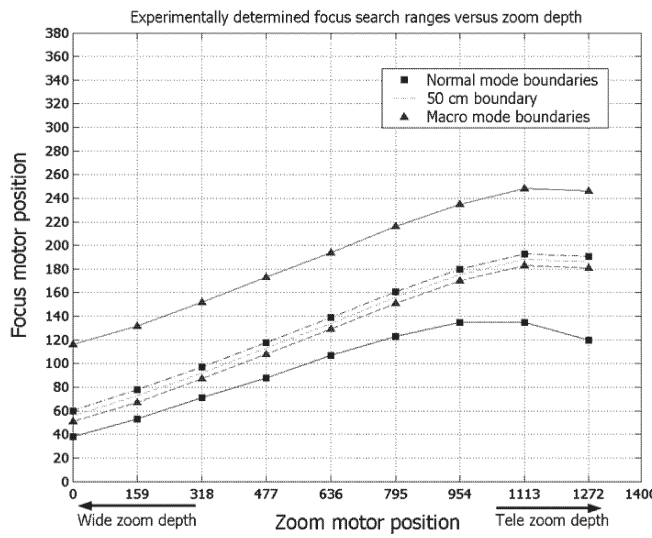

Figure 11. Experimentally determined reduced focus search ranges for the standard lens module.

AF Lag Time Reduction

Following the discussion above, strategies were developed to address the three major factors affecting the AF lag time, namely number of steps within the focus search range for a given zoom depth, step size increments used to control the movement of the stepper motor by the RS procedure, and synchronization of the processes tying together the entire AF algorithm.

Focus Search Range Reduction. The number of steps to be searched within the search range is directly proportional to the time needed to move the stepper motor during the first stage of the developed AF algorithm. So it is desired to reduce the focus search range as much as possible. Due to the optical relationship between in-focus positions and zoom depths, it was not necessary to search through the entire 371 step motor positions in order to determine an in-focus position for an object at an arbitrary distance away from the camera. Thus, in order to limit the focus search ranges, an experiment was done to determine the search range boundaries for the nine available zoom depths by noting the absolute in-focus step motor position for an object at infinity and for an object closer to the camera. It was noticed that the distance to the camera ensuring a valid in-focus position for a closer object increased as the zoom depth was increased from the wide angle towards the telephoto angle. The topmost curve and the bottommost curve in Fig. 11 show the search range boundaries determined by this experiment. The bottommost curve was obtained by recording the in-focus step motor positions for each zoom depth for an object at infinity (approximately 7 meters). For the topmost curve, the object was moved closer to the camera and the corresponding in-focus step motor position was recorded. This process was repeated until the camera could no longer bring the object into focus at which point the previously obtained in-focus step motor position was used as the nearest focusing step motor position. It should be noted that this experimentation is general purpose in the sense that it can be carried out for any lens module.

An additional reduction in the focus search range was attained by further splitting each zoom focus search range into a Normal and a Macro mode, as mentioned before, and illustrated in Fig. 11. The boundary between Normal and Macro mode was chosen to be for an object at a distance of 50 cm from the camera. A survey of the contemporary digital cameras revealed that 50 cm was the most commonly used boundary between Normal and Macro mode.

Clearly, in general, AF lag time depends on this boundary. For example, when increasing this distance, AF lag time in Normal mode decreases due to a decrease in the focus search range, whereas in Macro mode, AF lag time increases due to an increase in the focus search range. Hence, normal mode was used for focusing of objects from infinity to 50 cm, while Macro mode was used for focusing of objects below 50 cm. Some overlap was permitted between the focus search ranges for Normal and Macro modes as a transition region to ensure that the camera did not lose focus when objects were placed within the overlapping ranges.

Variable Step-Size Increments. After reducing focus search ranges, an additional modification was made to the RS in order to reduce the number of sharpness values taken during the first stage of the AF algorithm. This was done in an effort to reduce the AF lag time while at the same time ensuring that any focusing offset was within the acceptable limit of focusing quality. This approach is analogous to the strategy of adjusting step sizes previously mentioned.

In order to accomplish this objective, the concept of variable Mid and Coarse step size increments was introduced noting that the step sizes for Mid and Coarse in the original RS were fixed to 3 steps and 5 steps, respectively. Since it was desired to keep the number of sharpness samples to a minimum, the Fine step size increment was excluded so that only variable Mid and Coarse searches were carried out before and after passing an in-focus position.

Unlike fixed step size increments, variable step size increments start with an initial step size, and the step size is incremented in single steps until the specified maximum limit is reached. In the Mid search, the step size is incremented as long as there is no fall in the sharpness function, at which point the search is switched to Coarse. On the other hand, in the Coarse search, the step size is incremented until there is no rise in the sharpness function, at which point the search is switched to Mid.

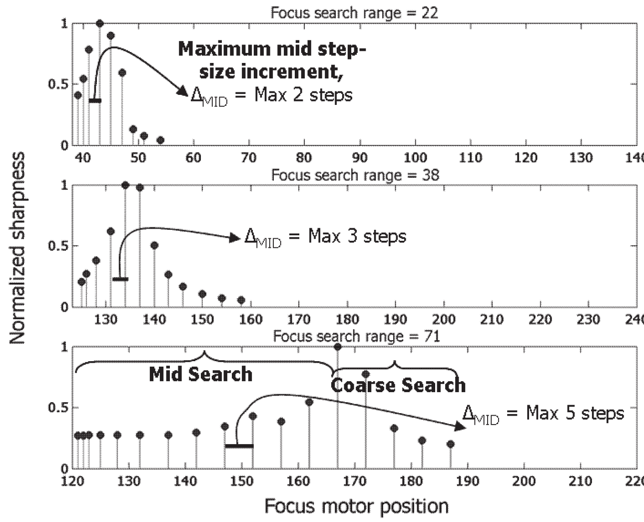

Figure 12. Differences in variable step sizes depending on size of the focus search range.

The maximum limit of variable step size increments depends on the zoom depth and thus is a function of the size of the focus search range. As illustrated in Fig. 11, zoom depths corresponding to wider angles have smaller focus search ranges while zoom depths corresponding to telephoto angles have larger focus search ranges. Therefore, smaller focus search ranges should not be searched very coarsely, since only a few steps need to be searched to find an appropriate in-focus position. On the other hand, large focus search ranges should not be searched finely, since this would add a significant amount of time to AF lag time.

Experimentation was performed to determine the maximum limit of variable step size increments in order to achieve the greatest possible reduction in the AF lag time, while maintaining a low offset with respect to the true focus position as determined by the GS. Figure 12 depicts how the maximum limit of variable step size increments changes based on the zoom depth. For zoom depths having lower focus search ranges, the maximum step size increment was limited to 2 or 3 steps since even an offset of 2 steps from the true focus position resulted in a noticeable loss in sharpness due to the relatively few search positions. Similarly, for the zoom depths having higher focus search ranges, the maximum step size increment was limited to 5 steps in order to reduce the AF lag time while maintaining a low offset from the true focus position. The use of variable step size increments provided a mechanism to reduce the AF lag time while maintaining an offset of less than 3 steps from the true focus position, which caused no visually noticeable loss in sharpness quality. Table I provides the experimentally determined limits of the variable step size increments for both Mid and Coarse searches as a function of the zoom depth and its respective focus search range. Note that the adjustment approach discussed here is general purpose in the sense that it can be applied to other lens modules.

TABLE I. Variable Step Size Limits Based on Number of Steps in a Focus Search Range

Search range	Mid step size limits		Coarse step size limits	
	ΔInitial	ΔMaximum	ΔInitial	ΔMaximum
(steps)	(steps)	(steps)	(steps)	(steps)
0 — 30	1	2	2	3
30 — 50	1	3	3	4
> 50	1	5	5	6

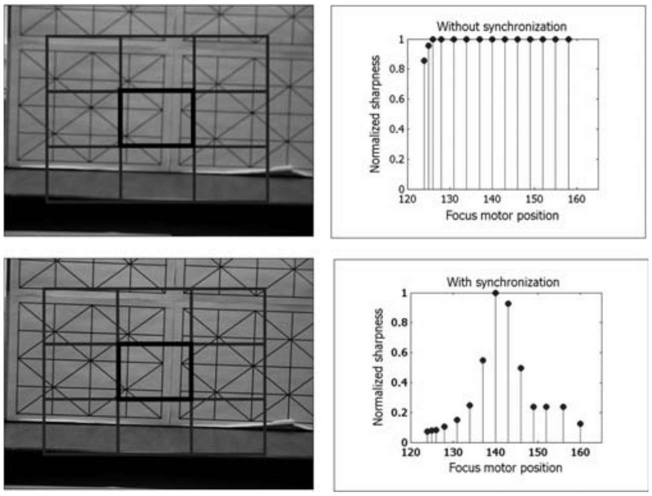

Figure 13. Effect of timing synchronization on in-focus position accuracy.

Synchronization of Events. In general, the process of AF consists of the movement of the focus stepper motor by a specified number of step positions as decided by the modified rule-based search. After the motor has been moved to the desired position, some delay time is needed to expose the image sensor to the current focus motor position and to let the AF engine process the image. After exposure, the sharpness sample for the current focus motor position can then be calculated. The sequence repeats as the focus motor is driven by the modified rule-based search procedure over the focus search range.

The crucial factor regarding synchronization is the time delay needed after the movement of the motor in order to have a proper exposure of the image sensor and to let the AF engine process the image data. Due to the use of variable step sizes, it was observed that a different amount of time delay was needed to wait for the sharpness calculation to be completed by the AF engine depending on the number of step positions moved by the focus motor. One method of dealing with this real-time issue was to actually vary the time delay according to the step size increment, but this led to an excessive increase in the AF lag time.

Instead of varying the time delay, another approach was considered based on the hardware feature available. Since two SDRAM buffers are utilized to store the AF statistics provided by the AF engine, the AF engine updates the AF statistics corresponding to the current step position in one buffer, while the other buffer holds the data for the previous step position. By using the buffer holding the AF data for the previous motor position during the current iteration, the extra delay to wait for the AF engine to calculate the statistics for the current step position was avoided, resulting in a reduction in the AF lag time.

TABLE II. Performance Summary for the Standard AF Algorithm on DM270 with Standard Lens

Global Search (method)	AF Lag Time (seconds)	AF Speed (controls)	AF Power (steps)
Normal Mode	1.420	38.4	72.4
Macro Mode	2.394	67.0	129.4

TABLE III. Performance Summary for the Developed AF Algorithm on DM270 with Standard Lens

Modified rule-based (method)	AF Accuracy (offset steps)	AF Lag Time (seconds)	AF Speed (controls)	AF Power (steps)
Normal Mode	1.0	0.635	16.6	71.6
Macro Mode	1.3	0.730	19.0	126.7

In addition to affecting the AF lag time, this synchronization improved the in-focus position accuracy as illustrated in Fig. 13. As shown in this figure, when no time delay was used, the sharpness calculation outcome from the AF engine was not always reliable. There needed to be at least one vertical synchronization wait from the time the motor stopped moving and began the next iteration of the search procedure.

Experimental Results and Discussion

The performance of the developed passive AF algorithm incorporating the modifications to support multiple focusing regions and the steps taken to reduce the AF lag time was compared with the standard global-search algorithm in terms of in-focus position accuracy, AF lag time and power consumption. The deviation or offset of the obtained in-focus position from the true focus position provides a measure of in-focus position accuracy. AF lag time denotes the total amount of time the AF algorithm takes to determine an appropriate in-focus position within an appropriate focus search range illustrated in Fig. 11, while the total number of steps moved by the focus motor is considered to be a measure of power consumption.

To provide a quantitative comparison, extensive testing was carried out by capturing various images under different indoor and outdoor lighting conditions, in both Normal and Macro modes, for a variety of objects of interest. A total of 276 images in Normal mode and 48 images in Macro mode were captured over different zoom depths. It should be mentioned that before the AF algorithm was executed, the focus step motor was moved from its current position to the corresponding home or starting position for the set zoom depth and focus mode (Normal or Macro) as shown in Fig. 11. In this figure, the bottom Normal boundary curve provides the home positions for Normal mode, while the bottom Macro boundary curve provides the home positions for Macro mode.

Tables II and III summarize the performance of the standard AF algorithm and the developed AF algorithm, respectively. The results show that the developed AF algorithm is fast and accurate with an average offset of no more than 1.3 steps from the true focus position, an average AF lag time of less than 0.735 seconds, and a power consumption comparable to the standard algorithm. It should be noted that the offset of 1.3 steps from the true focus position resulted in only a 0.7% gray level mean-square error or in no visually noticeable loss in sharpness. Basically, it was observed that 90% of the scenes produced an AF lag time of less than 0.75 seconds and 95% of the scenes an offset of less than 3 steps from the true focus position with no visually noticeable loss in sharpness.

Conclusion

In this article, several key real-time issues, often overlooked in the existing literature but faced by those developing contrast sensing passive AF algorithms for digital still cameras, were thoroughly addressed in an effort to assist AF designers to make well informed choices. Specifically, each of the design choices among the three main components of a passive AF algorithm, namely focusing region, sharpness measure and search procedure were discussed in terms of how the features of an available hardware platform limits these choices. In order to solidify the concepts presented, the choices behind the development and implementation of a developed passive AF algorithm on a digital still camera processor were also presented. The developed AF algorithm utilizes a modified rule-based search procedure which uses multiple focusing regions and supports multimodal sharpness functions. Extensive testing was performed with varied photographic compositions to assess the AF performance in terms of in-focus position accuracy, AF lag time, and power consumption. It was shown that the developed AF algorithm is more than 50% faster, accurate with no noticeable loss in sharpness quality and with no additional power consumption over the standard AF algorithm. Finally, it should be realized that further reduction in the AF lag time can be achieved by using a higher quality lens module having a faster physical speed.

Acknowledgment. The authors wish to thank the Imaging and Audio Business Unit of Texas Instruments for their support of this work.

References

1. Texas Instruments Inc., *DSC marketing research*, http://www.ti.com (2004).
2. K. Illgner, H.-G. Gruber, P. Gelabert, J. Liang, Y. Yoo, W. Rabadi, and R. Talluri, Programmable DSP platform for digital still cameras, *Proc. IEEE Conf. Acoustics, Speech, and Signal Processing*, **4,** 2235 (1999).
3. M. Mancuso and S. Battiato, An introduction to the digital still camera technology, *ST J. System Research* **2**(2), 1 (2001).
4. N. Nathaniel, N.P. Aun, and A. Marcelo, Practical issues in pixel-based autofocusing for machine vision, *Proc. IEEE Conf. Robotics and Automation* **3,** 2791 (2001).
5. K.-C. Choi, J.-S. Lee and S.-J. Ko, New autofocusing technique using the frequency selective weighted median filter for video cameras, *IEEE Trans. Consumer Electronics* **45**(3), 820 (1999).
6. J.-S. Lee, Y.Y. Jung, B.-S. Kim, and S.-J. Ko, An advanced video camera system with robust AF, AE, and AWB control, *IEEE Trans. Consumer Electronics* **47**(3), 694 (2001).
7. J. He, R. Zhou and Z. Hong, Modified fast climbing search autofocus algorithm with adaptive step size searching technique for digital camera, *IEEE Trans. Consumer Electronics* **49**(2), 257 (2003).
8. M. Subbarao and J.-K. Tyan, The optimal focus measure for passive autofocusing and depth-from-focus, *Proc. SPIE* **2598,** 89 (1995).
9. Y.-P. Tan, B. Thomas and T. Acharya, Method and apparatus for automatic focusing in an image capture system using symmetric FIR filters, US Patent 6,373,481 to Intel Corp. (2002).
10. N. Kehtarnavaz and H.-J. Oh, Development and real-time implementation of a rule-based auto-focus algorithm, *Real-Time Imaging* **9**(3), 197 (2003).
11. T. Yeh, N. Sampat, S. Venkataraman, and R. Kremens, System implications of implementing auto-focus on consumer digital cameras, *Proc. IS&T PICs Conference*, IS&T, Springfield, VA, 1999, p. 44.
12. Canon Inc., Canon PowerShot S45 Digital Still Camera Manual (2002).
13. F. Groen, I. Young and G. Ligthart, A comparison of different focus functions for use in autofocus algorithms, *Cytometry*, **39**(1), 1 (2000).
14. A. Santos, C. O. de Solorzana, J. J. Vaquero, J. M. Pena, N. Malipica, and F. del Pozo, Evaluation of autofocus functions in molecular cytogenic analysis, *J. Microscopy* **188,** 264 (1997).
15. M. Subbarao, T. Choi and A. Nikzad, Focusing techniques, *J. Opt. Eng.* **32**(11), 2824 (1993).
16. J-H. Lee, K.-S. Kim, B.-D. Nam, J.-C. Lee, Y.-M. Kwon, and H.-G. Kim, Implementation of a passive automatic focusing algorithm for digital still camera, *IEEE Trans. Consumer Electronics* **41**(3), 449 (1995).
17. K. L. Chan, A. Chang and J. Lin, Auto-focusing algorithms in programmable digital cameras, http://ise.stanford.edu/class/psych221/projects/99/klchan/index.html (1999).
18. M. Gamadia, V. Peddigari, N. Kehtarnavaz, S.-Y. Lee, and G. Cook, Real-time implementation of auto focus on the TI DSC processor, *Proc. SPIE* **5297,** 10 (2004).
19. M. V. Shirvaikar, An optimal measure for camera focus and exposure, *Proc. 36th IEEE Southeastern Symposium on System Theory*, IEEE Press, Los Alminitos, CA, 2004, p. 472.
20. Texas Instruments Inc., *TMS320DM270 CPU and Peripherals Technical Reference Manual* (2003).