McGill University

HIGH-RESOLUTION VIDEO SYNTHESIS FROM MIXED-RESOLUTION VIDEO BASED ON THE ESTIMATE-AND-CORRECT METHOD

by

Stéphane Pelletier

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of **Master of Engineering**.

in

Electrical and Computer Engineering

Department of Electrical and Computer Engineering

Montreal, Canada

March 2003

Your file Votre référence
ISBN: 0-612-88379-5
Our file Notre référence
ISBN: 0-612-88379-5

Abstract

A technique to increase the frame rate of digital video cameras at high-resolution is presented. The method relies on special video hardware capable of simultaneously generating low-speed, high-resolution frames and high-speed, low-resolution frames. The algorithm follows an estimate-and-correct approach, in which a high-resolution estimate is first produced by translating the pixels of the high-resolution frames produced by the camera with respect to the motion dynamic observed in the low-resolution ones. The estimate is then compared against the current low-resolution frame and corrected locally as necessary for consistency with the latter. This is done by replacing the wrong pixels of the estimate with pixels from a bilinear interpolation of the current low-resolution frame. Because of their longer exposure time, high-resolution frames are more prone to motion blur than low-resolution frames, so a motion blur reduction step is also applied. Simulations demonstrate the ability of our technique in synthesizing high-quality, high-resolution frames at modest computational expense.

Résumé

Une méthode destinée à augmenter la fréquence des images produites par les caméras numériques à haute résolution est présentée. La technique utilise un dispositif spécial capable de produire simultanément des images de haute-résolution à basse fréquence d'échantillonnage, ainsi que des images de basse résolution à haute fréquence d'échantillonnage. L'algorithme adopte une approche appelée estimation-et-correction, dans laquelle un estimé de haute résolution est d'abord produit en déplaçant les pixels des images de haute résolution en fonction des déplacements observés dans les images de basse résolution. L'estimé est ensuite comparé à l'image de basse résolution courante et corrigé localement afin d'être conforme à cette dernière. La correction est effectuée en remplaçant les mauvais pixels dans l'estimé par ceux d'une interpolation bilinéaire de l'image de basse-résolution courante. En raison de leur temps d'exposition plus long, les images de haute-résolution sont plus sensibles au flou causé par le mouvement d'objets dans la scène que les images de basse résolution. Par conséquent, une étape de réduction des effets de flou causés par les mouvements d'objets est appliquée aux images de haute résolution. Des simulations démontrent l'efficacité de notre technique pour produire des images de haute qualité à haut débit.

Acknowledgements

I would like to thank my supervisor, Dr. Jeremy R. Cooperstock, for his support and encouragement during my master degree. Through his guidance, he managed to give me the perfect balance between providing direction and encouraging independence. He also gave me the opportunity to get familiar with research and to discover a field I like. His advice and editorial help in the preparation of this thesis were very appreciated. I wish to thank Stephen Spackman for providing me with the main idea behind this thesis. He often gave me helpful suggestions and comments for low-level optimizations, as well as inspiration for my work. Also a special thank to Yin Jianfeng for helping me starting with LaTeX and to Deniz Sarikaya for her technical support with Linux. I am also grateful to Charles Roy, for assistance with proofreading. Finally, none of this would have been possible without the support of my parents, Raymond Pelletier and Ghislaine Drapeau, to whom I dedicate this thesis.

Contents

List of Figures

Chapter 1

Introduction

1.1 Problem description

In computer vision applications whose performance depends on the level of detail and the temporal accuracy of video, there is always a demand for better video quality. In order to generate a video frame, imaging devices accumulate photons over a 2D matrix of light sensors, whose number determines the maximum achievable resolution of the camera [1]. The exposure (or integration) time of a single frame must be chosen so that each such sensor receives a sufficient number of photons to allow for a statistically accurate measure of the light intensity at its location. This is partly dependent on the surface area of the sensor. A physically smaller element requires a proportionately longer exposure time to produce a usable image, which, in turn, determines the maximum frame rate that can be achieved by a camera; shorter exposure times allow the video device to produce frames at a higher rate. One approach to reducing the exposure time is to increase the size of the lens in order to focus a greater number of photons onto the matrix of light sensors. However, this entails increasing the physical size of the camera and is not well suited for applications requiring near-field focus, as this may result in image distortion. Another solution is to employ an auxiliary light source to illuminate the scene. This may be limited only to certain applications where additional illumination is both feasible and acceptable.

To obtain high-resolution video at higher frame rates than that allowed by the integration time of light sensors, we propose using special video hardware capable of simultane-

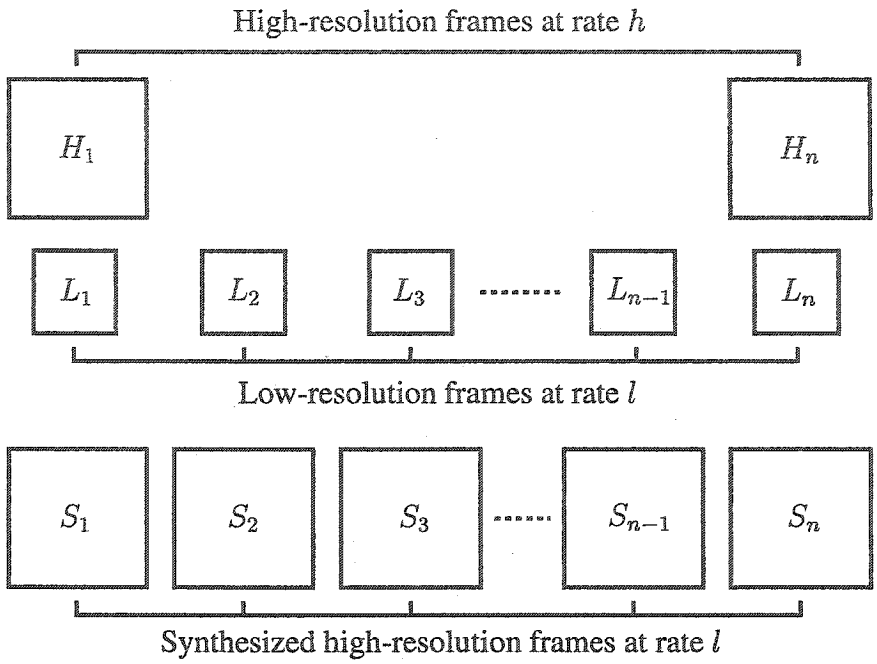


Figure 1.1. The specialized video hardware simultaneously produces two video sequences of the same scene; one high-resolution sequence at frame rate h and one low-resolution sequence at frame rate l. An image processing algorithm is then used to synthesize high-speed high-resolution frames from these two sequences.

ously generating high-resolution frames H at frame rate h and low-resolution frames L at a frame rate l, as depicted in Figure 1.1.

The high- and low-resolution frames represent the same scene and are used respectively to capture the high-frequency details of the scene and the motion of objects in the scene. The idea is to apply an image-processing algorithm to both sequences of frames H and L in order to synthesize a high-resolution video sequence S at high frame rate l having the detail level of the high-resolution frames H and the motion dynamic of the low-resolution frames L. The high-resolution frame rate h should ideally correspond to the highest frame rate allowed by the sensor physics at high-resolution. Since low-resolution frames involve the accumulation of incident photons over a larger sensor surface for each pixel and, thus, require less time to integrate than high resolution frames, we would expect the frame rate l of the low-resolution sequence to be higher than h. Hereafter, n will refer to the number of low-resolution frames captured for every high-resolution frame captured by the video hardware.

1.2 Literature Review

Descriptions of other mixed-resolution video synthesis techniques are scarce. This is likely due to the fact that this category of algorithm requires a device that produces high- and low-resolution frames in the manner described in Figure 1.1. In constrast, a great amount of work has been performed on a related problem, that of synthesizing high-resolution images from a set of low-quality, low-resolution ones by using the redundant information contained in the latter. The techniques used to solve this problem belong to the category of *super-resolution techniques*. Another category of methods consists of using a single low-resolution image to produce an image of better visual quality; we will refer to these methods as *single image enhancement techniques*. In the following paragraphs, we will summarize the state of the art of these categories of methods.

1.2.1 Single Image Enhancement Techniques

The reduction of blur and noise levels in an image, which is known as single image restoration, is a well-known problem in image processing theory. Three techniques are generally used to obtain practical restoration algorithms: the maximum likelihood (ML) estimator, the maximum *a posteriori* (MAP) probability estimator and the projection onto convex sets (POCS) [2][3][4][5][6]. Single image restoration is an ill-posed inverse problem [7], because many solutions exist given an input image. One method to tackle this problem consists of constraining the solution by employing *a priori* knowledge concerning the form of the solution, which can be provided by constraints such as local smoothness, edge preservation, positivity and energy boundedness.

Another method to enhance the visual quality of an image is interpolation, a technique often used inside digital cameras to increase their resolution artificially. The interpolation problem consists of reconstructing an ideal image from an undersampled version. Since the value of a pixel is obtained by accumulating photons over a surface, a low-resolution image cannot be considered as a perfectly undersampled version of an ideal image, although it is

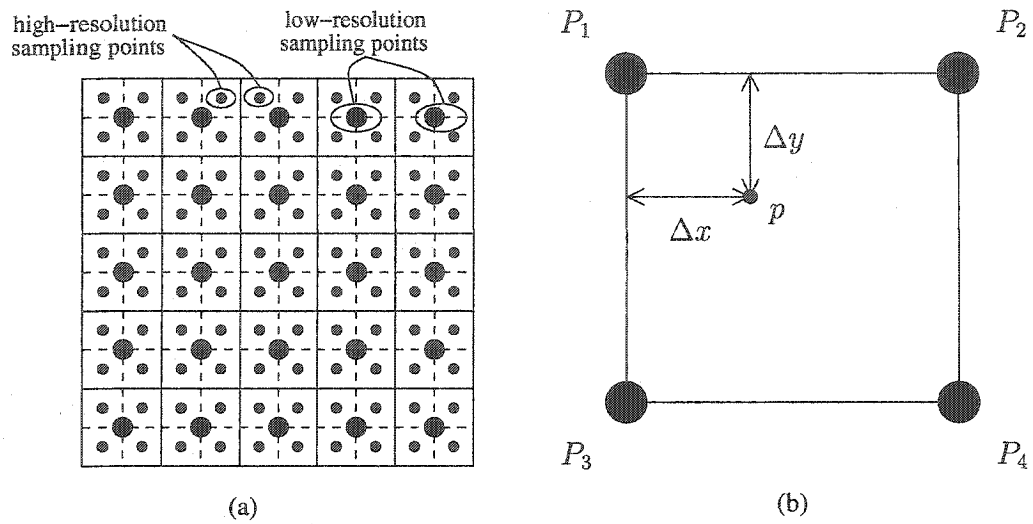


Figure 1.2. (a) Given a low-resolution sampling of an image (big dots), one wants to resample it at high-resolution (small dots). (b) The value of a point on the high-resolution grid can be interpolated by using its four adjacent low-resolution sampling points.

often convenient to assume that this is the case. As shown in Figure 1.2(a), the value of a low-resolution pixel can be thought of as the value of the ideal image at the location corresponding to the center of this pixel (big dots). In order to produce a high-resolution frame, one wants to determine the values of the ideal image at the locations corresponding to the center of the high-resolution pixels (small dots).

Different interpolation techniques exist, such as bilinear interpolation [8] and bicubic interpolation [9]. For the four points P_1 through P_4, illustrated in Figure 1.2(b), equation 1.1 explains how to calculate the value of the high-resolution pixel p using bilinear interpolation.

$$\begin{aligned} p = & P_1 \times (1 - \Delta x) \times (1 - \Delta y) + \\ & P_2 \times \Delta x \times (1 - \Delta y) + \\ & P_3 \times (1 - \Delta x) \times \Delta y + \\ & P_4 \times \Delta x \times \Delta y \end{aligned} \tag{1.1}$$

In the particular case of color images, it is possible to use the information provided by the three RGB components to obtain redundant information of the scene, which can then be used to improve the resolution of the image. Such an approach is described by Zomet and

Peleg [21]. In general, single image enhancement methods require minimal computational load and memory.

1.2.2 Super-Resolution Techniques

The enhancement of computer technology has made possible the development of more sophisticated and more computationally intensive algorithms, able to synthesize a high-resolution image from a sequence of degraded, undersampled ones [10][11][12][13][14]. A significant amount of research has been carried out in super-resolution techniques and a review of existing methods is presented by Borman *et al* [15]. These can essentially be divided into frequency domain [10][11][12] and spatial domain methods [13][14][16]; the former tend to be simpler and are preferred for applications involving global translation motion. Spatial domain methods, however, are better suited for general video sequences that may contain local motion.

The advantage of using many images is that it provides additional novel observation data, which comes from the fact that each image contains similar, but not identical information of the scene; in most super-resolution applications, each undersampled image has either a different blur level, most often obtained by changing the focus of the camera while taking the image sequence, or a different displacement with respect to the scene, either because of camera or object motion during the capture.

Keren, Peleg and Brada [17] presented a two-step spatial domain approach to super-resolution reconstruction. The first step consists of evaluating the displacement between the different low-resolution frames and creating a temporary image made of non-uniformly spaced samples, as shown in Figure 1.3. The process of evaluating the displacement between images is called *registration*. In Keren *et al*'s method, the displacement between images is assumed to be global; that is, if local motion occurs within the scene, registration will not produce accurate results. Once registration is complete, the samples are interpolated on a high-resolution grid to produce an estimate of the high-resolution frame. The

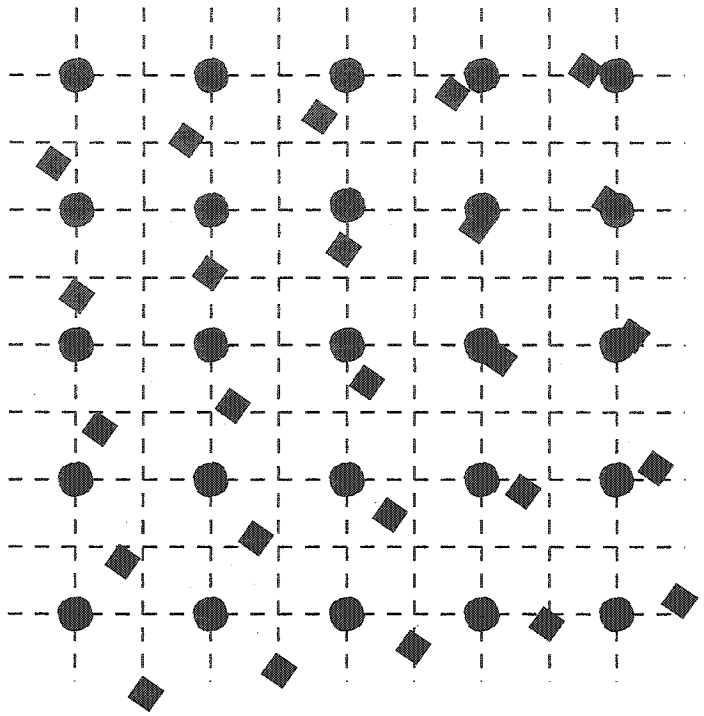


Figure 1.3. Two low-resolution point sets are mapped to a high-resolution grid. The alignment of the first point set (circles) is not the same as that of the second point set (diamonds). The information provided by both sets can be used to interpolate the value at the center of each high-resolution pixel.

second step is an iterative process, belonging to the class of simulate-and-correct methods. First, an imaging process is applied to the high-resolution estimate. This consists of different degradation kernels that simulate the effect of resolution reduction by the camera. The result of this step is a sequence of *simulated* low-resolution frames, which are then compared to the observed low-resolution frames. The high-resolution estimate is modified to reduce the error between the simulated and the observed low-resolution images. The modifications are performed using a *backprojection kernel*, which scales the correction to each high-resolution pixel according to its contribution to the error in the simulated images. This simulate-and-correct step is then repeated and stops when the error between the simulated and observed images drops below a certain threshold, or after a given number of iterations. Irani *et al* [18] extended this work to deal with local motion.

Elad and Feuer [16] proposed a technique that relates to adaptive filtering theory. Their method consists of using a time and space filter that operates on a set of low-resolution

images to produce a restored sequence of images at higher resolution. This is applied using either the LMS or the RLS algorithm [19] [20] and the operation complexity is proportional to the number of pixels in the frames. The restoration procedure consists of solving a large set of sparse linear equations. Similar to the methods of Keren and Irani, Elad and Feuer's technique depends on a prior evaluation of motion between low-resolution frames.

In general, the success of super-resolution techniques strongly relies on their capability to evaluate accurately the displacements between low-resolution images. Therefore, the accuracy of the registration step is often the limiting factor in super-resolution reconstruction performance [13].

1.2.3 Mixed-Resolution Input Techniques

At least one recent reference [22] is available on the topic of mixed-resolution techniques and presents a digital camera that meets the high frame rate and resolution requirements of the movie industry. The motivation of this work is to reduce the amount of video data when bandwidth and/or storage capacity are limited.

The technique uses a special camera with a beamsplitter that sends incoming light from the scene along two paths; one leads to three high-resolution CCD image sensors, one for each color component, and the other leads to a low-resolution CCD image sensor. The three high-resolution imagers produce color frames at a low frame rate, whereas the low-resolution imager produces grayscale frames at a high frame rate. Image processing electronics are used to generate high-resolution frames at high-speed from these two inputs. When a high-resolution frame is captured by the high-resolution sensor, no work needs to be done; otherwise, the algorithm is applied.

The image processing algorithm is a relatively straightforward application of technology similar to video compression [23][24][25][26]. A motion vector field describing the motion of objects within the scene from the previous low-resolution frame to the current one is computed. Then, a high-resolution frame is generated by moving the pixels of the

previous high-resolution frame according to the motion field computed at low-resolution. This prediction step is done only with the pixels for which the motion from the last frame to the current frame is known. In particular, when pixels are disoccluded by moving objects, information from the low-resolution frames is used to interpolate high-resolution pixels. However, as the low-resolution frames do not provide color information, the quality of the reconstruction in these zones is limited. Thus, the pixels around the transitions between moving objects and the background are treated differently from other areas of the scene. The identification of the transition regions is enhanced by finding the boundary of objects. This last step, which is basically an edge detection algorithm, is performed using a high pass filter.

1.3 Overview of Our Work

This thesis presents a technique for building a digital camera capable of generating video at a higher frame rate than that allowed by standard digital cameras. As discussed earlier, physics of the imaging sensors imposes a limit to the maximum speed at which a digital camera can capture frames. Under similar light conditions, low-resolution sensors can generate video at a higher frame rate than can be achieved by high-resolution sensors. The purpose of the presented technique is to generate high-resolution frames at the low-resolution frame rate, which, hereafter, will be referred to as the *high* frame rate. Since this value depends on the scene illumination and the size of the camera lens, it may in fact be quite low. Therefore, we should note that the term *high* frame rate is meant only to distinguish it from the correspondingly *low* rate associated with high-resolution frames.

In the presented technique, high-resolution information is retrieved directly from the few high-resolution frames generated by the video hardware, which is more efficient than extracting this information from several undersampled frames. The algorithm uses the estimate-and-correct method, whose main advantages are simplicity and computational efficiency, this allowing for real-time applications. In particular, the method includes a mo-

tion evaluation step, which, in part because of the wide distribution of MPEG video encoding applications, may be performed by readily available specialized hardware for motion estimation [27][28][29]. Although our method uses the same kind of motion evaluation algorithm as that found in many video compression techniques such as MPEG, the purpose of our algorithm is entirely different. Whereas MPEG and its variants provide compression of a sequence of video images, our algorithm aims to enhance image quality and frame rate. Therefore, comparisons between our algorithm and video compression methods are inappropriate and will not be considered in this dissertation.

The algorithm requires a video device generating simultaneously high- and low-resolution frames as illustrated in Figure 1.1. Since the exposure time of the low-resolution frames is shorter, these frames have a better temporal accuracy than the high-resolution frames. Conversely, the high-resolution frames have a better spatial accuracy. The technique efficiently interpolates high-resolution frames in between those generated by the video hardware through a compromise between temporal and spatial accuracy.

A fast mixed-resolution block matching algorithm is used to evaluate coarsely pixel motion between the last interpolated (synthesized) high-resolution frame S_{t-1} and the current low-resolution frame L_t generated by the camera. The technique takes advantage of a subsequent correction process to reduce the computational cost of the motion evaluation step without excessively degrading the quality of the synthesized frames. The result is an interpolated frame that contains mostly high-resolution and some low-resolution features. The low-resolution features, which are smoothed by simple frame interpolation, tend to appear when abrupt motion occurs in the scene. Because of the longer exposure time required to generate the high-resolution frames, these are more sensitive to motion blur than the low-resolution frames. For this reason, the algorithm includes a step to reduce the effects of motion blur to a level equivalent to that of low-resolution frames.

Simulation results indicate that the visual quality of the frames synthesized by the algorithm is far better than that of the corresponding low-resolution frames. In particular, the

quality of the frame areas corresponding to static parts of the scene is optimal, in the sense that they are equivalent in quality to the few high-resolution frames generated by the camera. The algorithm also performs well when reconstructing areas of the scene that involve slow and translation-based motion.

1.4 Layout of the Thesis

This thesis is organized as follows. Chapter 2 explains the frame acquisition model and the inherent problem of differing sensitivities to motion blur between the high- and low-resolution frames generated by our camera. The proposed digital camera architecture is also described in this chapter. Chapter 3 develops the technique used to synthesize high-resolution frames from mixed-resolution ones. Chapter 4 presents the simulation results obtained and discusses the performance of the proposed algorithm. Finally, some conclusions and remarks on future directions of this research are offered in Chapter 5

1.5 Contributions of Thesis

The main contributions of this thesis are the description of a special video device capable of generating high-resolution frames at a higher frame rate than that achievable with standard digital cameras and an algorithm for synthesizing these frames. This device could be useful in applications that impose size or bandwidth constraints on the camera, as well as those in which ambient light levels preclude high-resolution exposure given the limitations of sensor physics.

Chapter 2

Frame Acquisition Model

2.1 Frame Acquisition by a Special Video Camera

A digital camera produces a video sequence by capturing images of a scene at regular time intervals, which define the frame rate of the camera. The image receptor area of a digital imaging device is made up of a 2D array of light sensor elements called photosites. Each photosite converts incident light into photocurrent $i_{ph}(t)$, whose intensity gives the value of the corresponding pixel. However, as photocurrent is too small to be measured directly, a digital camera cannot instantaneously capture the content of a scene; instead, the photocurrent must be integrated onto a capacitor and the charge $Q(t)$ is read out after a given time period T, called the exposure time. The amount of charge accumulated in a photosite is a linear function of the incident illumination intensity and the integration period. If noise is discounted, the relation between $i_{ph}(t)$ and $Q(t)$ is given by:

$$i_{ph}(t) = \frac{dQ(t)}{dt} \tag{2.1}$$

Thus, assuming constant illumination during the exposure period, the intensity of the photocurrent at the beginning of the exposure can be estimated as follows:

$$i_{ph}(t) = \frac{Q(T)}{T} \tag{2.2}$$

Figure 2.1 illustrates the computation of $i_{ph}(t)$ from the accumulated charge $Q(T)$. The sensitivity of a photosite depends on the size of its light reception surface; a photosite with

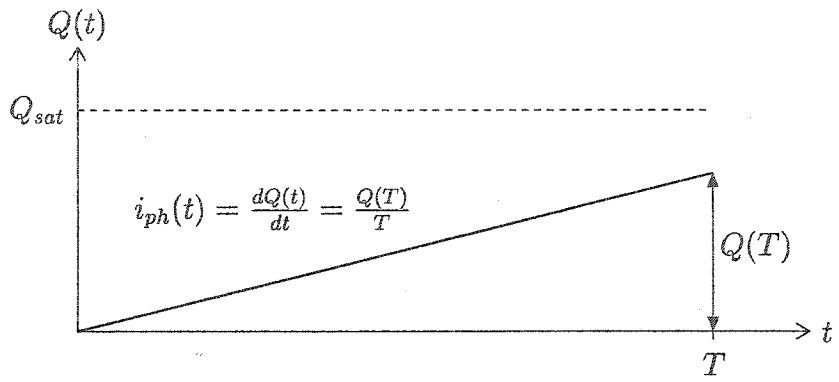


Figure 2.1. Assuming constant illumination during the exposure period, $\frac{Q(T)}{T}$ gives the slope of the charge accumulation function $Q(t)$ at any point t, which is, by definition, the photocurrent $i_{ph}(t)$.

a small surface is less sensitive to light and thus, requires more time to produce a pixel. As a result, the surface size of the photosites and the light intensity define the minimal exposure time T_{min} necessary to capture a frame. One should also note that the maximum charge, Q_{sat}, that a photosite can accumulate corresponds to its saturation level. If the exposure time is too long, all the pixels will take on the same value, corresponding to the maximum light intensity.

Hereafter, time $t = 0$ will refer to the beginning of a new exposure time. If the charge accumulated into a photosite is read after a shorter exposure time, for example, $t = T_{min}/4$, the value read will not be reliable. However, if the capacitor values for adjacent photosites are added by groups of four, which would correspond to reading values from a photosite whose reception surface is four times larger, the summed values will be sufficiently accurate to produce a reliable low-resolution frame. This technique is actually used in some multiresolution video devices [30] [31], which can be programmed to generate frames at different resolutions. These devices can increase their light sensitivity at the cost of resolution.

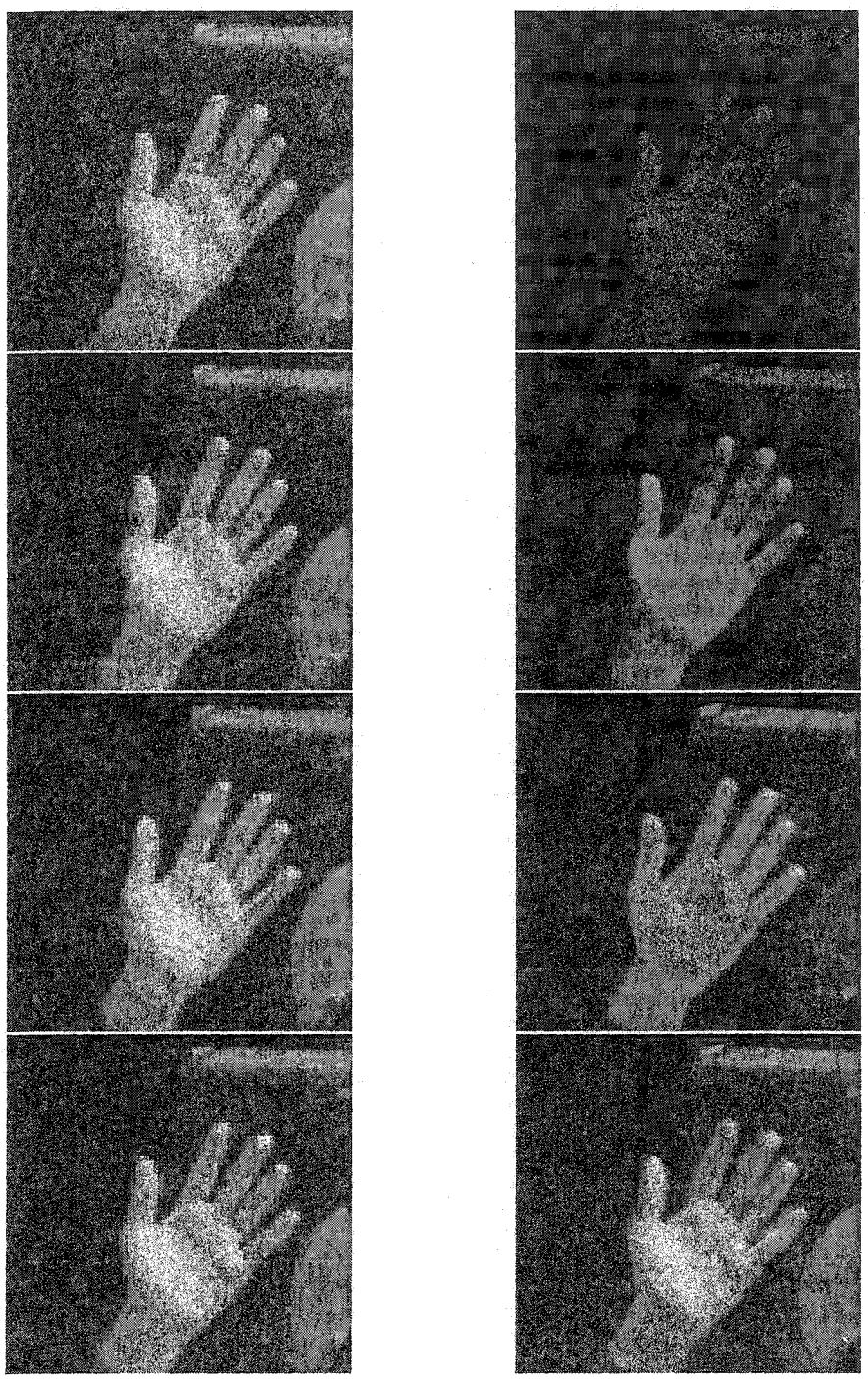
This technique can be expanded to construct a special camera capable of generating simultaneously low-resolution frames at high-speed and high-resolution frames at low-speed. By using appropriate circuitry, it is possible to generate a low-resolution frame every T_{min}/n and a high-resolution frame every T_{min} time period, where n gives the num-

ber of high-resolution pixels that are combined to produce a low-resolution pixel. For the purpose of illustration, we will continue to assume a value of $n = 4$ for subsequent discussion. As a result, the high frame rate will correspond to quadruple that of the original high-resolution frames captured by the camera.

2.2 Motion Blur Sensitivity

An object moving within the scene during the exposure time spreads its light information over many photosites, which produces a motion blur effect. Obviously, the longer the exposure time, the stronger the effect. As the exposure time required for the acquisition of a high-resolution frame with the camera described above is approximately n times that for a low-resolution one, the captured high-resolution frames are more sensitive to motion blur, as illustrated in Figure 2.2. The left column shows four low-resolution frames that were captured successively by the video camera, using an exposure time of $T_{min}/4$ for each, while the right column shows the acquisition process of the corresponding high-resolution frame at different stages, each corresponding to the end of the exposure of the low-resolution frame to its left. While motion blur in each low-resolution frame corresponds to an integration period of $t = T_{min}/4$, motion blur in the high-resolution frame corresponds to an integration period of T_{min}. This explains why the hand in the resulting high-resolution frame is blurred.

The purpose of the described technique is to synthesize the high-resolution frames that would be produced by a camera whose photosites are sufficiently sensitive to generate high-resolution frames within an exposure time of $T_{min}/4$. In other words, the motion blur inside the synthesized high-resolution frames should, ideally, correspond to this exposure time. As mentioned earlier, the technique synthesizes high-resolution frames by translating pixels of the few high-resolution frames generated by the video device. If these few high-resolution frames are affected by motion blur, the effect will be propagated to the synthesized frames; as a result, sensitivity to motion blur in the synthesized frames will



(a) (b)

Figure 2.2. Sensitivity to motion blur as a function of the exposure time. (a) Low-resolution frames obtained with an integration period of $t = T_{min}/4$. (b) Progressive integration state of a high-resolution frame obtained with an integration period of $t = T_{min}$.

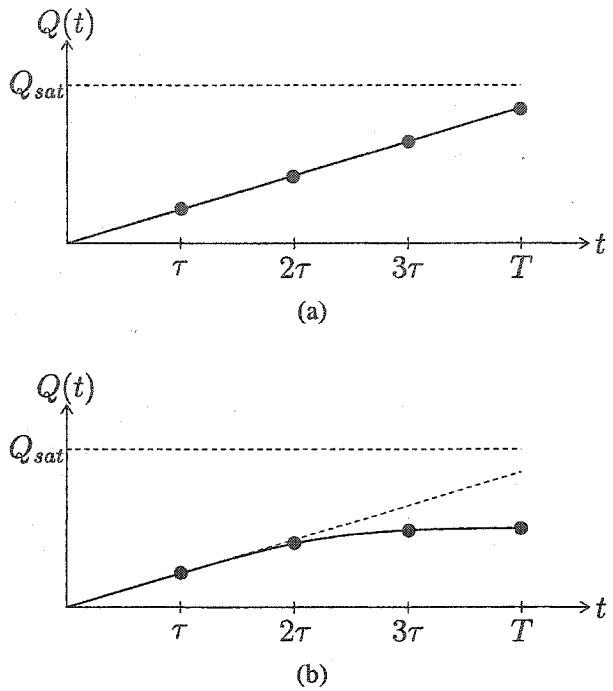
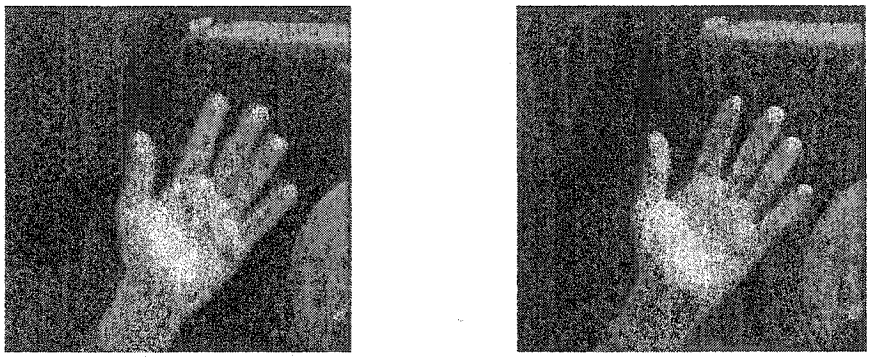


Figure 2.3. Effect of motion on the photocurrent. (a) Under conditions of no motion, the photocurrent generated by the sensor remains constant throughout the integration period. (b) With scene motion, photo current may vary during integration.

correspond to an exposure time of T_{min}, rather than $T_{min}/4$.

A technique to reduce the effects of motion blur in a frame is presented by Liu *et al* [32]. The technique consists of using special video hardware capable of capturing many images within a normal exposure time. Taking advantage of the extra information provided by these underexposed images, the motion blur level in the generated frames can be reduced. Figure 2.3(a) represents the accumulation of the charge in a photosite when light intensity does not vary during the exposure time; in this case, one can assume that no motion occurs, since the accumulation slope, i.e. the photocurrent, is constant. On the other hand, Figure 2.3(b) indicates the effect of motion on photocurrent. Under such condition, equation 2.2 does not provide an accurate measure of the photocurrent at the beginning of the exposure time.

Instead, different values of $Q(t)$ measured during an exposure are used to estimate the photocurrent at the beginning of the exposure time $(t = 0)$. In fact, the method determines



(a) (b)

Figure 2.4. Motion blur reduction in a high-resolution frame using its underexposed states. (a) Frame affected by motion blur (b) Frame after motion blur reduction.

whether motion has occurred during the exposure time by examining the general shape of the integration curve. If motion is detected, the point τ_m corresponding to the beginning of motion is identified. Then, only the information from $t = 0$ to $t = \tau_m$ is used to estimate the photocurrent at the beginning of the exposure time. If no motion occurs, then all the points are used to produce the estimate. The number of points used to estimate the photocurrent has an influence on its estimate. For instance, if motion occurs early during the exposure time, the estimate will not be accurate. In our case, we are interested in the value of the photocurrent at $t = 3\tau$, as it corresponds to the beginning of the integration time of the ideal frame we wish to reconstruct. By using a similar approach to Liu *et al* [32], our technique can estimate this value and, thus, can reduce motion blur in the generated high-resolution frames, as shown in Figure 2.4.

2.3 Video Hardware Design

This section describes an imaging system capable of producing simultaneously low-speed, high-resolution frames and high-speed, low-resolution frames, using a special multi-resolution sensor. Since the underexposed high-resolution frames are required to reduce motion blur of the high-resolution frames produced, they must also be made available by the imaging system hardware.

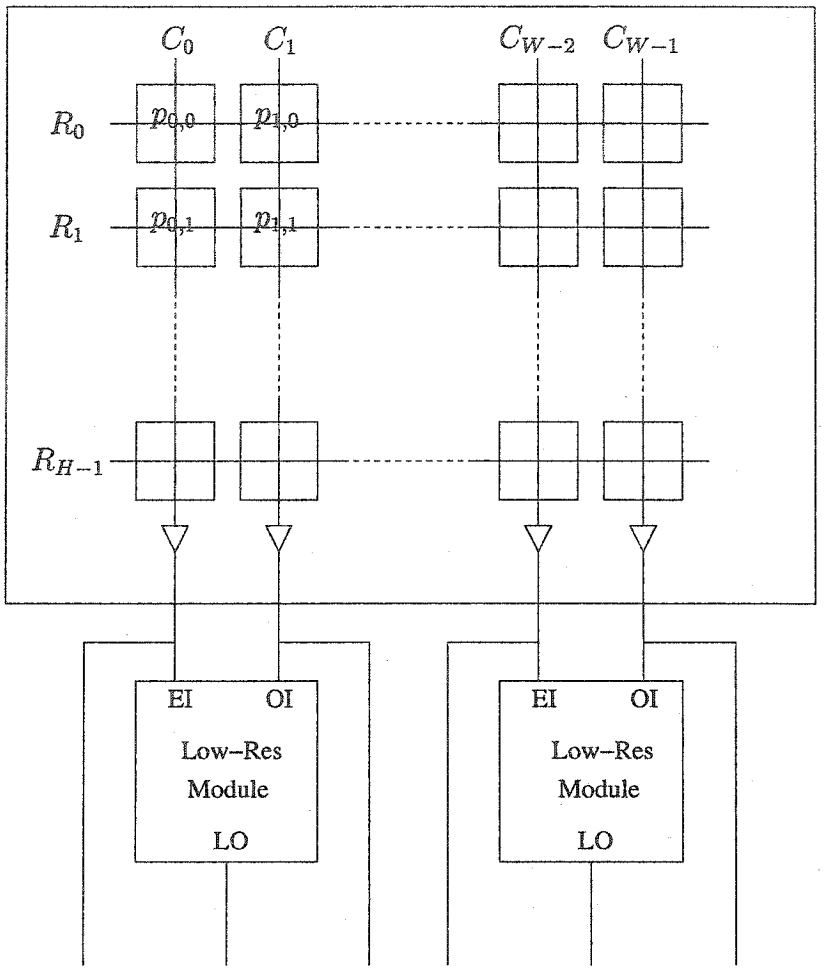


Figure 2.5. Global architecture for a grayscale mixed-resolution sensor.

As we assume a value of $n = 4$, the width and height ratios between low- and high-resolution frames generated are both two. Also, in order to simplify the figures and explanations, mixed-resolution hardware that generates grayscale frames will be presented. A color-equivalent device could be designed easily by replacing each single photosite by three, each one modulated by the appropriate RGB color filter. Figure 2.5 illustrates that the photosites are arranged in an array of H rows by W columns, from which each pixel P_{ij} can be read by enabling the column-line C_i and the row-line R_j. For efficiency, the parallel organization of the device permits the simultaneous read-out of the W pixels of the j^{th} row by enabling the row-line R_j and all the column-lines C_i $(0 \leq i < W)$. Thus, by successively enabling each row-line R_j $(0 \leq j < H)$, the entire array of photosites can be

read.

The exposure time of each frame is chosen to satisfy the minimum time requirements to obtain good results for the high-resolution frames. At the end of the exposure time, the photosites are reset and a new acquisition cycle begins. However, multiple[1] non-destructive readouts are performed within a single exposure time in order to generate a number of low-resolution frames. While recent advances in CMOS technology allow such operations [33], this could also be simulated by CCD technology using an auxiliary memory to save the accumulated values of the photosites and resetting the sensors after each readout.

The generation of the low-resolution frames is accomplished as follows. Pairs of column-lines are fed into low-resolution modules, illustrated in Figure 2.5. Each such module combines the underexposed high-resolution pixels from four adjacent photosites to produce a well-exposed low-resolution pixel. Since a column line can only carry one value at a time, it is not possible to transmit the four pixels simultaneously. Assuming that an entire row can be read during a clock cycle, two clock cycles are required to transmit four underexposed pixels to a low-resolution module. As a result, the first row of low-resolution pixels will be produced after the first two rows of high-resolution pixels have been received, and so forth for successive rows.

A low-resolution module is detailed in Figure 2.6. The low-resolution module receives and sums together the values of two adjacent high-resolution pixels, one coming from an even column (EI) and one from an odd column (OI). If these come from an even row, the write-enable (WE) signal of the memory register is set and the result of the addition is stored in a register, whereas if they come from an odd row, the result of the addition is added to the previously stored sum. This ensures that each low-resolution pixel obtains the sum of four adjacent high-resolution pixels, arranged in a 2x2 grid. The value of the low-resolution pixel produced is sent to the low-resolution output (LO). Note that since the photosites are not reset after each read operation, it is necessary to subtract the contribution of their

[1] In this example, four.

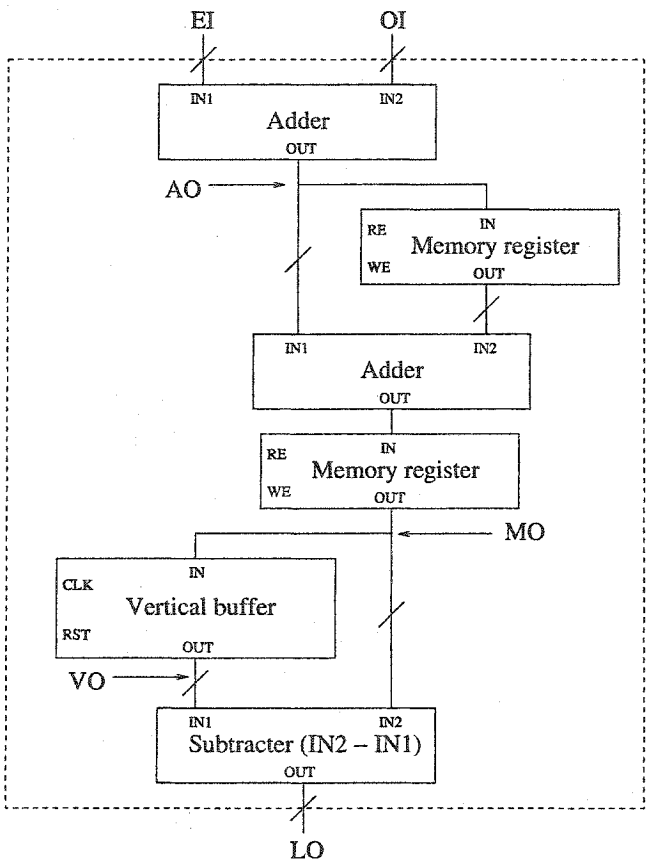


Figure 2.6. Possible implementation of a low-resolution module.

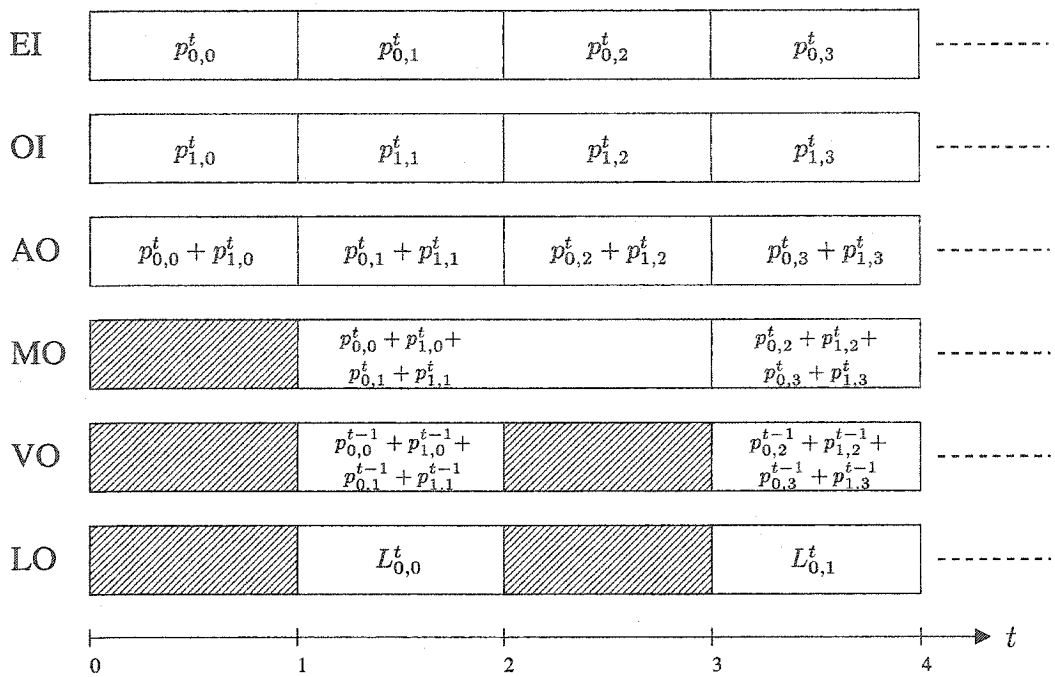


Figure 2.7. Timeline diagram for the low-resolution module connected to the first two columns of the sensor.

previous values from each low-resolution pixel produced by the module. Thus, an auxiliary memory, whose size is proportional to the number of low-resolution pixels, is employed to store these values. Figure 2.7 is a timeline diagram that illustrates the operation of the low-resolution module at the bottom-left of Figure 2.5, using the corresponding symbols from Figures 2.5 and 2.6. Each low-resolution pixel produced is identified as $L^t_{0,k}$, where k is its corresponding row in the low-resolution frame.

The vertical buffer is used to store the addition results corresponding to the $H/2$ low-resolution pixels of the column in the previous frame. A clock signal (CLK) permits sequential access to each memory location as each pixel in the column is processed. For each low-resolution pixel produced, the vertical buffer is used twice. First, its values are subtracted from the sum of the second adder corresponding to each low-resolution pixel being calculated. Next, this result is written back to the current memory location, replacing the previous sum.

The entire contents of the vertical buffer can be zeroed by the RST signal, which must be done at the beginning of every new exposure cycle of a high-resolution frame. Thus, the

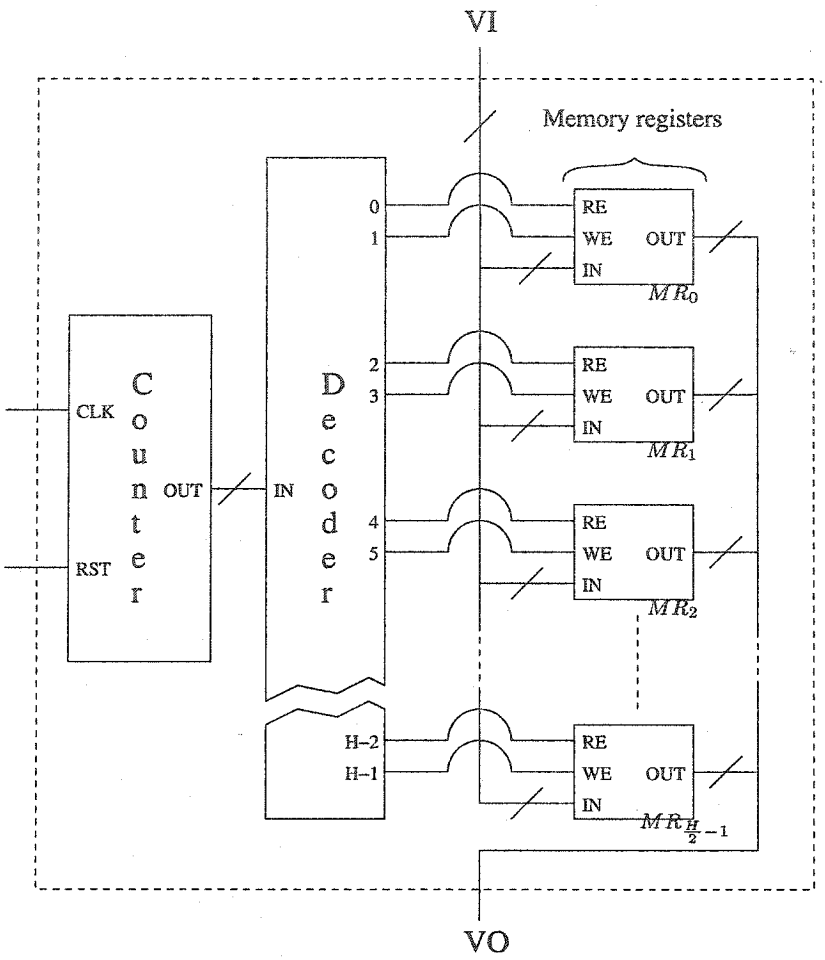


Figure 2.8. Possible implementation of a vertical buffer.

low-resolution pixels produced after the first non-destructive readout of an exposure cycle will not be changed by the subtraction operation, described above.

Figure 2.8 demonstrates a possible implementation of the vertical buffer. The reset signal (RST) resets the counter, whereas the clock signal (CLK) increments the counter. A decoder is used to enable one memory cell at a time, either in read or write mode. The i^{th} clock impulse, following the reset, will read enable the $(\frac{i}{2})^{th}$ memory cell if i is even or write enable the $(\frac{i-1}{2})^{th}$ cell if i is odd.

The decoder is a component whose outputs are all set to zero, except the one referenced by the value at the input, namely the counter output. Thus, by continually incrementing the counter, each output of the decoder will be set, one at a time, allowing individual access to

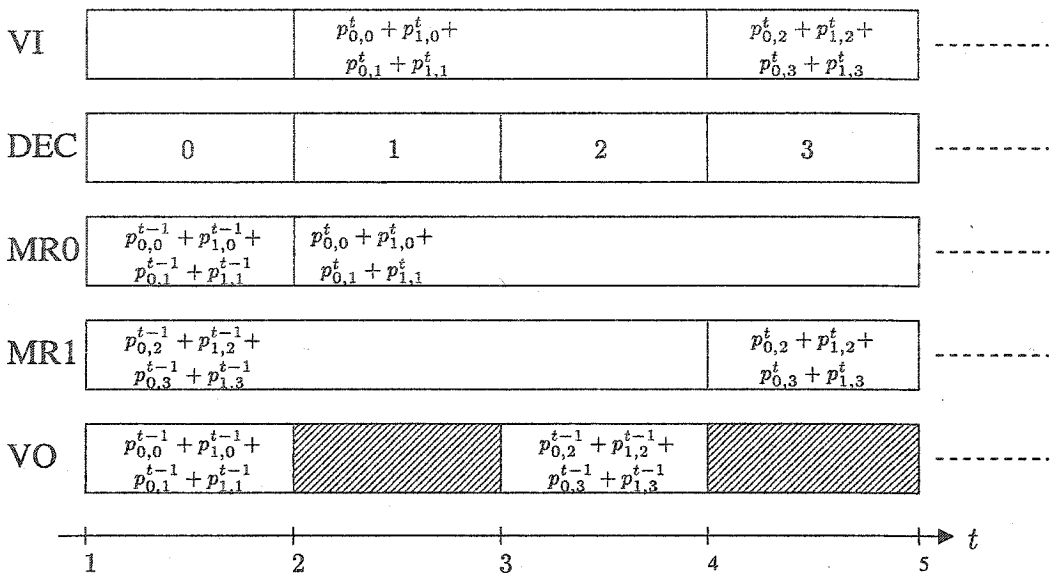


Figure 2.9. Timeline diagram for the vertical buffer inside the low-resolution module connected to the first two columns of the sensor. DEC identifies the active output of the decoder.

each memory register. Figure 2.9 is a timeline diagram that illustrates the operation of the vertical buffer.

Chapter 3

Estimate-and-Correct Method

3.1 Concept

Our strategy used to synthesize the current high-resolution frame S_t involves moving the pixels of the last frame S_{t-1} with respect to motion observed at low resolution. The problem with this approach is obvious: motion cues alone are insufficient to describe the scene changes from S_{t-1} to S_t. As an extreme example, if the scene contains a video display that changes from white to black, it would be impossible to move the white pixels in S_{t-1} to produce black ones in S_t. Therefore, the accuracy of the motion evaluation step depends on the nature of the scene changes; when objects move rapidly or when their motion cannot be expressed as a simple translation, it may be impossible to synthesize S_t from S_{t-1} alone. Furthermore, the computational expense of calculating motion generally increases with the complexity of the motion dynamics within the scene.

For these reasons, our method first produces a coarse estimate E_t of the high-resolution frame S_t by translating the pixels in S_{t-1} for which the motion dynamic can be computed efficiently at low resolution. A second step is then performed, which corrects the estimate E_t by patching it locally with low-resolution information. Thus, the algorithm can be divided into two steps, namely the high-resolution estimation and the estimate correction. The advantage of this approach is that it allows an efficient computation of the next frame S_t without expending an inordinate effort on areas of the scene that do not exhibit simple motion characteristics. The method actually performs a trade-off between temporal and

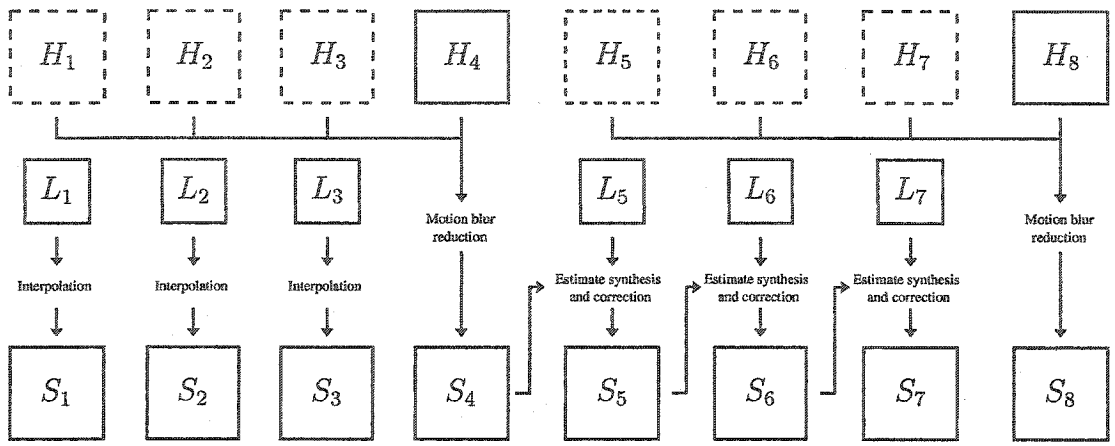


Figure 3.1. When the integration period of a high-resolution frame is completed, this frame and its three underexposed frames (dashed) are used to synthesize a high-resolution frame with less motion blur effect in it. In other cases, the estimate-and-correct method is applied between the current low-resolution frame and the last synthesized frame. However, the first three synthesized frames S are produced by bilinear interpolation from their corresponding low-resolution frame, as no high-resolution frame is available at the beginning.

spatial accuracy. That is, high-resolution information will be lost in areas where the motion evaluation is not obvious and thus cannot be computed quickly, for example in areas of disocclusion or at the edges of moving objects.

The frame rate, and in turn, the amount of scene change between frames are related directly to the speed of the algorithm. A slower algorithm results in a lower frame rate and thus, greater variations between frames. Reducing execution time allows us to achieve higher frame rates, which helps reduce scene changes and thus, improves the motion estimation step.

Figure 3.1 illustrates the application of our technique for $n = 4$, assuming that the video device was activated at time $t = 0$. As can be observed, the estimate-and-correct algorithm is not applied at every time step; specifically, when low-resolution frames L_1, L_2, and L_3 are captured, no high-resolution frame has yet been output by the video hardware. As a result, it is not possible to move the pixels of a previous high-resolution frame to produce the estimates E_1, E_2, and E_3. Instead, we use simple bilinear interpolation to synthesize the first three high-resolution frames S_1, S_2 and S_3. By $t = 4$, we have acquired sufficient data to synthesize S_4 by applying the motion blur reduction algorithm presented in Section

2.2. The estimate-and-correct algorithm is then used to synthesize the next three frames, S_5, S_6, and S_7.

More generally, the motion blur reduction algorithm is applied each time a high-resolution image is available. Otherwise, the estimate-and-correct method is applied, for which only the current low-resolution frame L_t and the previous synthesized frame S_{t-1} are used to synthesize S_t. It was mentioned earlier that the observation of motion within the scene is performed at low resolution, which could normally imply the utilization of both L_{t-1} and L_t to estimate motion from time $t-1$ to time t. However, as will be explained later, we use S_{t-1} instead of L_{t-1} for this step.

In observing Figure 3.1, one may wonder why the motion blur reduction strategy is not used to synthesize every high-resolution frame. This is because the estimation accuracy of photocurrent decreases in the presence of motion. Suppose a scene does not contain any moving objects from $t = 0$ to $t = 4$. In this case, the quality of the synthesized frame S_4 will be optimal. If an object starts moving at $t = 4$ and the motion blur reduction strategy is used to synthesize S_5, the quality of the latter will be inferior to that of S_4. However, capitalizing on the motion information observed from L_4 to L_5 to guide the translation of high resolution pixels from S_4 to S_5, we generally obtain improved results, as the frame S_4 is optimal.

3.2 High-Resolution Estimation

A widely used method of evaluating motion between two frames in a video sequence is the block matching algorithm (BMA). The BMA assumes that each area of the current frame can be obtained from the translation of some corresponding area in the previous frame. The technique usually consists of partitioning a frame F_t into equal-sized non-overlapping square blocks and finding, for each, the best matching block in the preceding frame F_{t-1}. The displacements of these best-matched blocks are represented as vectors, describing how the different parts of the scene moved from F_{t-1} to F_t. Because the representation of motion

of all the pixels in a block is reduced to a single motion vector, the evaluation computed by the BMA is necessarily coarse. This characteristic can cause a problem in some situations; for example, if a block contains the edge of an object that moves over a static background, the motion will be perceived as identical for both the object and the background within this block.

However, the BMA is quite efficient in terms of speed when compared to more complex motion estimation algorithms that allow nonrigid variations in their motion model [34]. Furthermore, in the small time intervals our algorithm expects between frames, it is likely that pure translation estimation will be sufficient to describe the majority of scene changes. For this reason, our technique uses BMA to perform motion evaluation.

The sum of squared-differences (SSD) may be used to measure the quality of match between a pattern block at position (x, y) in F_t and a candidate block at position $(x + u, y + v)$ in F_{t-1}.

$$\mathrm{SSD}_{(x,y)}(u,v) = \sum_{j=0}^{B-1}\sum_{i=0}^{B-1}\big(F_t(x+i,y+j) - F_{t-1}(x+u+i,y+v+j)\big)^2 \tag{3.1}$$

where $B \times B$ is the block size. The best matching block (u_b, v_b) in F_{t-1} is the candidate block that satisfies

$$(u_b, v_b) = \arg\min \mathrm{SSD}_{(x,y)}(u,v) \tag{3.2}$$

To evaluate motion at low resolution, one would normally apply the BMA between L_{t-1} and L_t. However, this would provide motion evaluation accuracy equivalent to one low-resolution pixel at best. Applying a low-resolution motion vector to S_{t-1} to produce the estimate E_t would result in a lack of accuracy of the estimate, which, in turn, would lead to a lower quality synthesized frame S_t. Hence, we desire the motion evaluation at low resolution to be as precise as one high-resolution pixel; in other words, if the resolution enhancement from the low- to the high-resolution frames is a factor of r, then the motion evaluation at low resolution must be accurate to within $\frac{1}{r}$ low-resolution pixels. To achieve

this subpixel precision, we use a mixed-resolution block matching algorithm (MRBMA) to perform the motion evaluation between the last synthesized frame S_{t-1} and the current low-resolution frame L_t. The advantage of MRBMA over simple BMA is that it permits a match of a low-resolution block on a high-resolution alignment, which provides greater precision in the motion evaluation.

Like the BMA, MRBMA partitions the current low-resolution frame L_t into square blocks and finds, for each, the best matching block in the previous frame S_{t-1}. However, as S_{t-1} is at higher resolution than L_t, the best matching block in S_{t-1} will be a high-resolution representation of the pattern block to be matched in L_t. Mathematically, the MRBMA consists of increasing the resolution of L_t to match that of S_{t-1}. To do this, we use a simple scaling function, described by:

$$L_t^s(x, y) = L_t\big(\lfloor x/r \rfloor, \lfloor y/r \rfloor\big) \tag{3.3}$$

We may then apply a standard BMA algorithm between S_{t-1} and L_t^s using equation 3.1.

The MRBMA is the most critical component of the high-resolution video synthesis algorithm. Its efficiency is imperative to reduce execution time and its accuracy determines the quality of the synthesized high-resolution frames. While the full search BMA provides optimal results because it inspects every possible block within a search window, its computational cost is prohibitive for real-time applications. Therefore, a considerable amount of work has been done on the development of fast block-matching algorithms [35].

One way to accelerate the BMA consists of restricting the search area depending on the motion field; thus, the number of potential candidates to match for each pattern block is reduced. Although this approach does not guarantee a best match for each block, it can significantly accelerate the motion estimation step. Furthermore, in the case of our video system, trying to match a candidate block whose relative position is far from the current pattern block does not improve the quality of the estimate. That is, if significant changes due to rapid motion occur within a low-resolution exposure time, both the low- and high-resolution frames produced by the video hardware will suffer from motion blur.

As explained in Chapter 2, the motion blur reduction operation will reduce this effect in the high-resolution frame to a level comparable to that of low-resolution frames. If the motion level is too high, the difference in quality between a blur-reduced high-resolution frame and a bilinearly interpolated version of the low-resolution frame will not be significant, as both will appear smeared. Thus, there is no purpose in performing motion evaluation when rapid motion occurs. The MRBMA builds on methods that are designed to provide optimal results with short execution time under conditions of limited scene motion, and interrupts the search if a good match is not easily found.

To improve the efficiency of the MRBMA, we use the results from adjacent blocks as predictors. When calculating a motion vector for a given pattern block, the vectors corresponding to its neighbors, if known, are used to initialize the search window. The motivation for this optimization is obvious: adjacent blocks are likely to be part of the same object, and thus, tend to exhibit similar motion characteristics.

The method, illustrated in Figure 3.2, works as follows. For each pattern block in L_t^s, its best candidate block in S_{t-1} is to be found. Without loss of generality we can assume that the best candidate blocks B_1 and B_3, corresponding respectively to pattern blocks P_1 and P_3, have already been identified. Since P_2 is between P_1 and P_3 in L_t^s, one would expect to find its best match B_2 somewhere between B_1 and B_3 in S_{t-1}. However, in this particular case, P_1 and P_3 come from different regions in the preceding frame; that is, B_1 and B_3 are quite far from each other. This situation might correspond to a case in which two objects are approaching each other. Given this information, P_2 could be part of the same object as P_1 or P_3, or possibly both if the region contains pixels belonging to both objects. For this reason, the search process for B_2 expands from two different locations in S_{t-1}, as shown in Figure 3.2(d), and stops when either a reasonable match is found or it exceeds specified bounds on the search window.

Once the motion vector from S_{t-1} to L_t^s is evaluated, it is applied to S_{t-1} to produce E_t. Figure 3.3 shows an example of the estimation step, in which a foot is moving toward

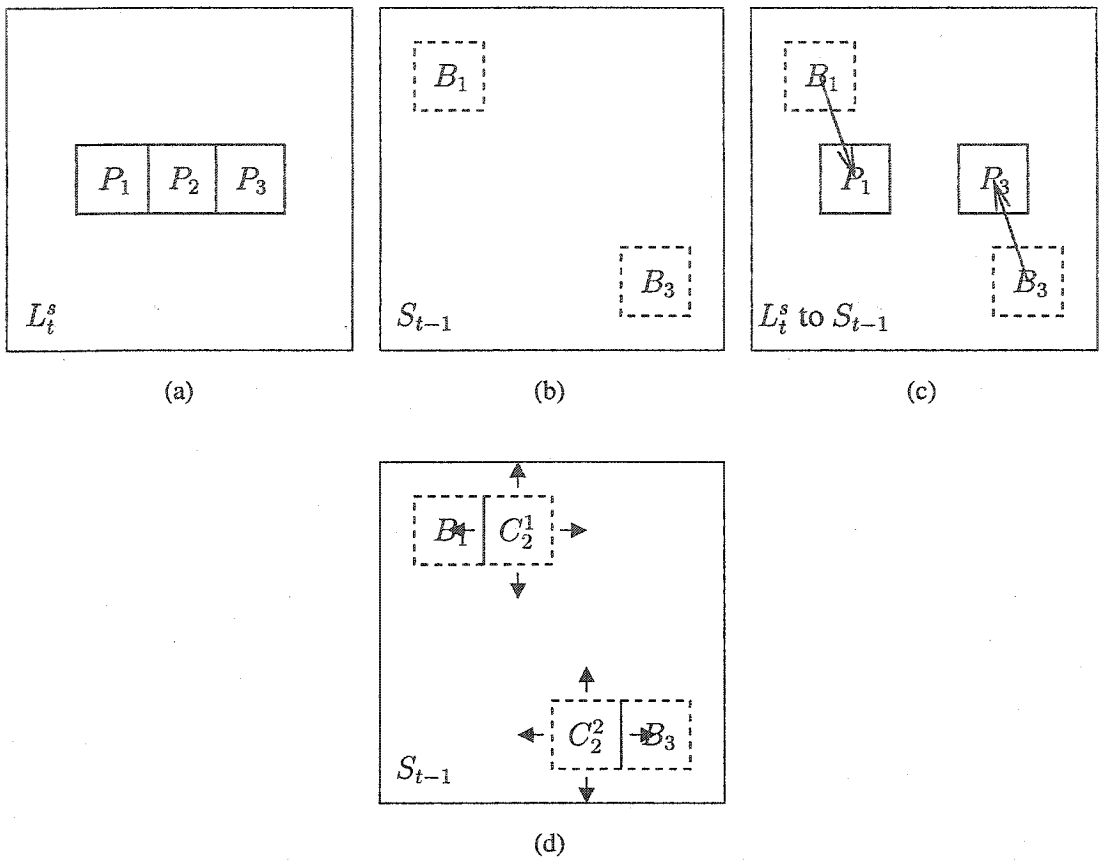


Figure 3.2. The motion vectors are to be computed between two frames. (a) P_1, P_2, and P_3 are adjacent blocks in L_t^s. (b) We assume that the best candidate blocks in S_{t-1} for P_1 and P_3 have already been found and are identified as B_1 and B_3 respectively. (c) Motion vectors for P_1 and P_3. (d) Given the relative positions of the three pattern blocks in L_t^s and the positions of B_1 and B_3 in S_{t-1}, two locations are predicted for B_2 in S_{t-1} and are identified as C_2^1 and C_2^2. The search then expands from these as needed to find the best match.

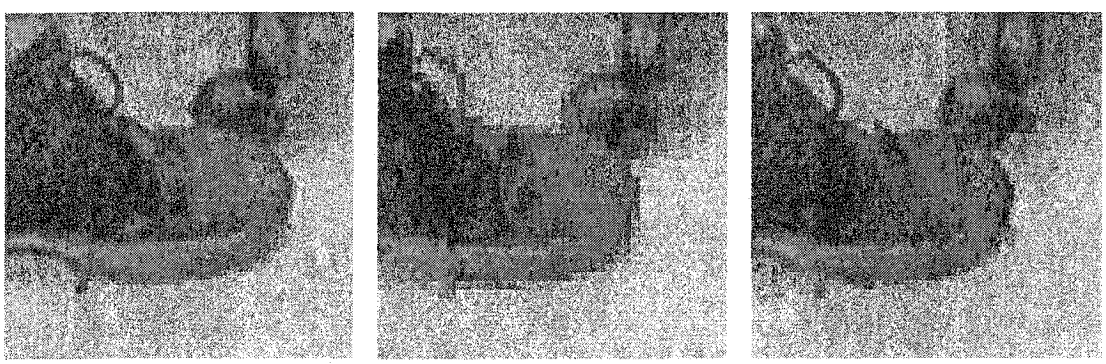


Figure 3.3. Estimate synthesis by translation of blocks. (a) The last synthesized frame S_{t-1}. (b) The current low-resolution frame L_t. (c) The motion vectors from S_{t-1} to L_t are computed with the MRBMA, and the motion dynamic is applied to S_{t-1} to produce E_t.

the left in front of the wheel of a chair. The first image corresponds to the high-resolution frame that was synthesized at time $t-1$, the middle image is the current low-resolution frame L_t, and the right image corresponds to the estimate E_t, obtained by applying to S_{t-1} the motion dynamic observed between S_{t-1} and L_t.

3.3 Estimate Correction

The estimate E_t will typically contain artifacts similar to those produced by a poor MPEG codec, mainly due to the fact that it is not always possible to find an adequate match in S_{t-1} for each pattern block in L_t. For example, observing the estimate in Figure 3.3, one notes that the lower part of the wheel has been moved with the end of the boot, as it was part of the same block in the BMA. These artifacts can be interpreted as a lack of temporal accuracy in the estimate, i.e., some pixels in the estimate do not correctly reflect the current state of the scene.

In order to reduce the visual effect of such artifacts, the associated pixels are corrected by introducing some information from a bilinearly interpolated high-resolution version B_t of the current low-resolution frame L_t. Although the interpolated version contains less detail than E_t, it is more accurate temporally, as it has been produced from the current low-resolution frame L_t, and generally of better visual quality than L_t itself.

Therefore, the synthesis of the high-resolution frame S_t actually consists of merging the estimate E_t and an interpolated version of L_t. The idea is to give more importance to E_t when it is deemed to be an accurate representation of the current state of the scene and less importance otherwise. A simple way to verify whether an arbitrary high-resolution image represents the same scene as a corresponding reference image at low-resolution is to subsample and compare it with the latter. Thus, the relative weight given to E_t and the interpolated version of L_t is based on the quality of match between the current low-resolution frame L_t and a subsampled version E'_t of the current high-resolution estimate

E_t, which, for a given location (x, y) can be expressed by the following equation:

$$M(x,y) = \frac{[L_t(x,y) - E'_t(x,y)]^2}{K^2} \tag{3.4}$$

where K is a chosen constant, based on the maximum possible color component value, to scale the values of $M(x,y)$ to $[0,1]$. For example, if the pixel depth of the generated images is eight bits, K should be set to 255, as this is the maximum possible value for the difference between any two pixels. If the result of the comparison is close to zero, the estimate is deemed to be a reasonable approximation of the ideal high-resolution frame at that location and thus a greater weight is given to it in the reconstruction process. Conversely, if the squared difference is high, then a greater weight is given to the current low-resolution image.

Experimental results suggest that the contribution of E_t in the synthesis process should be set to zero when the value given by equation 3.4 is above a given threshold t_{match}. Consequently, the merging process can be expressed by the following equation

$$\begin{aligned} S_t(x,y) =& \big(1 - D(x,y)\big) E_t(x,y) \\ &+ D(x,y) B_t(x,y) \end{aligned} \tag{3.5}$$

where

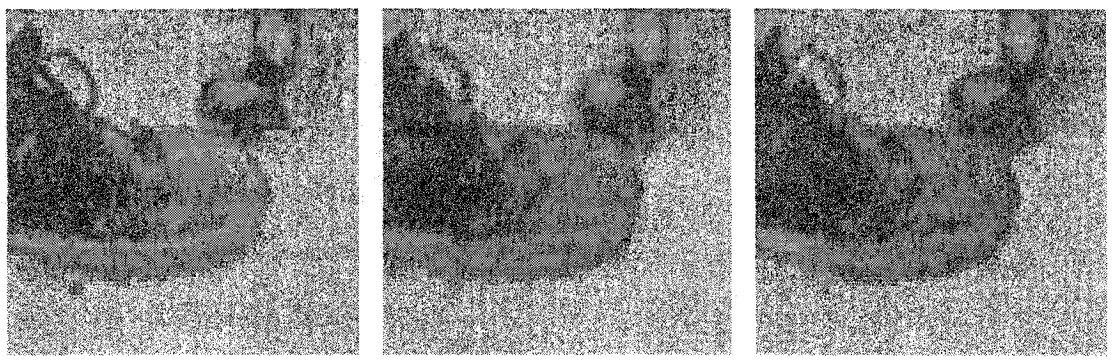
$$D(x,y) = \begin{cases} M(x,y) & \text{if } M(x,y) < t_{match}, \\ 1 & \text{otherwise.} \end{cases} \tag{3.6}$$

As such, estimate correction involves a trade-off between temporal and spatial accuracy; substituting low-resolution information in the region of an estimation error resolves temporal problems but reduces spatial accuracy. Figure 3.4 illustrates the result of the estimate correction step when applied to the example of Figure 3.3.

3.4 Comparison to other Methods

3.4.1 Super-Resolution Techniques

Our estimate-and-correct method follows the same philosophy as the simulate-and-correct method used by Keren *et al* [17] and Irani *et al* [18]. That is, a high-resolution estimate is



(a) (b) (c)

Figure 3.4. Estimate correction step. (a) High-resolution estimate E_t. (b) L_t (figure 3.3(b)) is bilinearly interpolated to produce B_t. (c) S_t is synthesized using equations 3.5 and 3.6, with $t_{match} = 0.0006$.

first produced and compared to the low-resolution observations. Corrections to the estimate are then performed to constrain the synthesized high-resolution frame to correspond to the observed data. The major difference lies in the way that these two steps are performed.

In their methods, the high-resolution estimate is obtained by interpolating many low-resolution frames, whereas in our case, the high-resolution estimate is obtained by moving the pixels of the previous synthesized frame S_{t-1} with respect to the motion observed from S_{t-1} to L_t. Also, rather than requiring a costly, iterative modification process, our correction method produces each high-resolution frame in a single step by performing a trade-off between two images.

Another difference is the constraint on the input data. In order to obtain good results with super-resolution techniques, the low-resolution frames must contain similar but different information, provided either by motion in the scene or motion of the camera itself. This restriction does not apply in our case, as the high-resolution information is provided directly by the occasional high-resolution frames captured.

3.4.2 Mixed-Resolution Input Techniques

Because the camera architecture proposed by Turner *et al* [22] uses distinct imaging sensors to produce the high- and low-resolution frames, the incoming light must be split between

each of these sensors. This beam splitting operation results in a reduction of the number of photons available to each sensor, and therefore increases the minimal exposure time necessary to obtain usable images. Unlike the previous camera, our architecture uses a single imaging sensor to produce both high- and low-resolution frames, reusing, in a sense, the same photons to produce both sequences, and thus permitting the capture of frames at a higher frequency. In this case, the same photons are used to produce both sequences of frames, which permits the capture of frames at a higher frame rate.

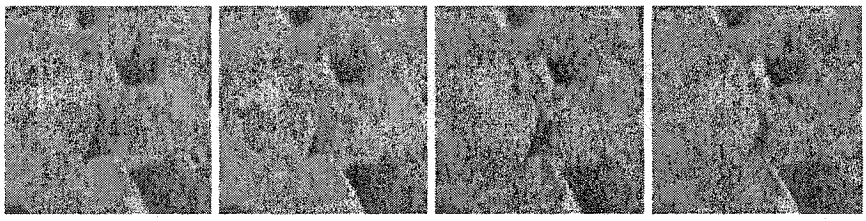
While both our method and Turner *et al* [22] move the pixels of the last synthesized high-resolution frame to produce the current one, the main difference between these approaches is how we address the problem that the information needed to generate S_t cannot always be found in S_{t-1}. Turner *et al* perform a global analysis of the low-resolution frames to define the transitions between moving objects and the background. In particular, the zones of disocclusion are identified and used to synthesize the high-resolution frame. Although this step is not explained in their patent description, it is clear that the quality improvement achieved is limited by the nature of the grayscale low-resolution frames. In our case, the high-resolution estimate is compared against the corresponding low-resolution frame, and the value of this comparison is used to determine locally whether or not the pixels in the estimate should be corrected. Furthermore, as our low-resolution frames contain color, the improvement achieved is potentially more significant. Finally, our method post-processes the high-resolution frames to reduce the effects of motion blur, a treatment not considered by Turner *et al.*

Chapter 4

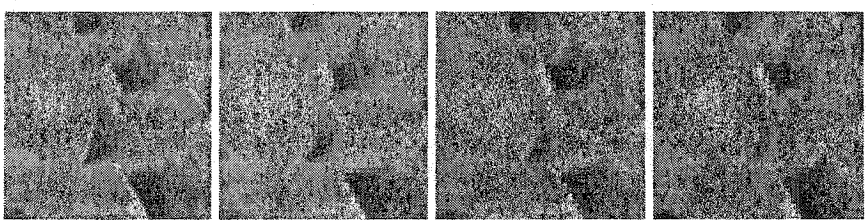
Experimental Results

To validate the current approach to high-resolution video synthesis, bilinear and bicubic interpolation algorithms were implemented. Next, both qualitatitve and quantitative comparisons were performed between our method and the others. For the purpose of a quantitative comparison, we used the average SSD error per pixel, measured between the images produced by the various methods and that of the corresponding ideal images. Since the mixed-resolution video hardware does not yet exist, its output had to be simulated. A sequence of ideal high-resolution frames were captured using a conventional fixed resolution camera. Low-resolution frames, as well as underexposed, blurred high-resolution frames were then generated from these ideal frames. The former were produced by subsampling, whereas the latter were simulated by degrading the ideal frames and mixing them as necessary to reproduce motion blur effects. The mixture of high- and low-resolution images was then provided to the algorithm as if it were a live sequence from a mixed-resolution video camera.

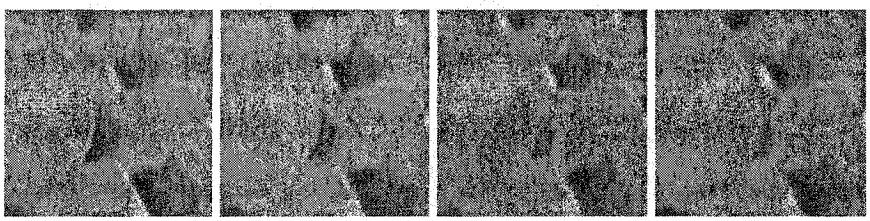
Figure 4.1 illustrates the high-resolution frames synthesized by the algorithm when applied to a short sequence of mixed-resolution frames, as well as the results obtained with bilinear and bicubic interpolation techniques. For comparison purposes, the ideal high-resolution frames, which represent a close-up of a thumb moving over a keyboard, are also presented. The first synthesized frame was obtained using the motion blur reduction algorithm described previously. In this example, a significant improvement in quality of



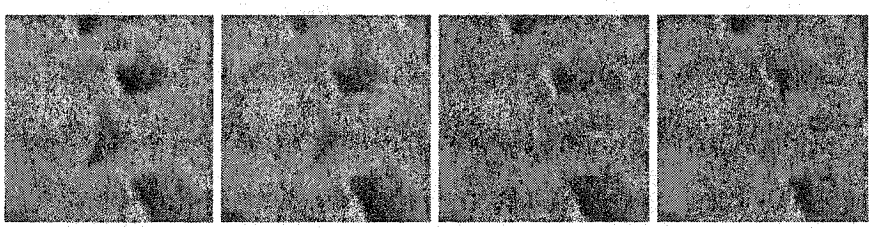
(a) Ideal high-resolution frames



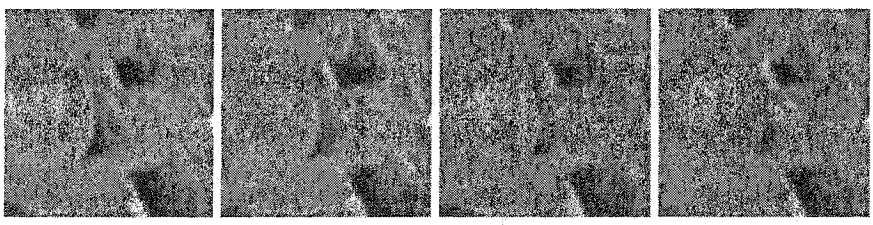
(b) Actual input to algorithm



(c) Frames synthesized by our method, avg. SSD = 14.3



(d) Frames synthesized by bilinear interpolation, avg. SSD = 13.6



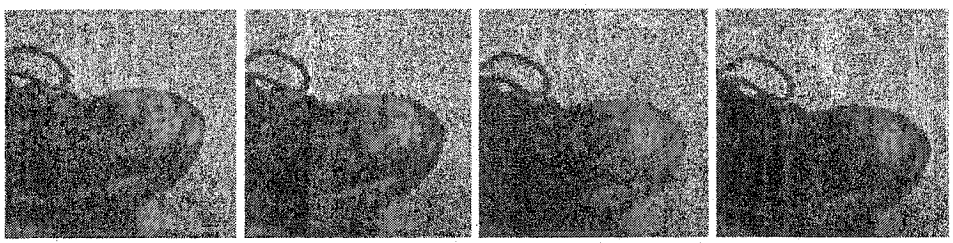
(e) Frames synthesized by bicubic interpolation, avg. SSD = 18.7

Figure 4.1. Comparison of the simulate-and-correct algorithm with bilinear and bicubic interpolation techniques. In this example, $n = 4$, and the size of the blocks in the motion estimation step is 8×8 low-resolution pixels. SSD is measured on a per pixel basis.

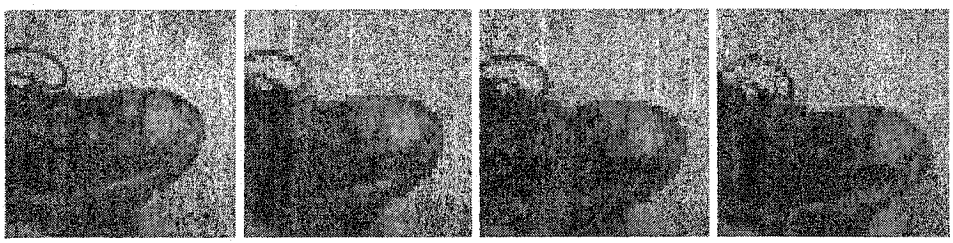
the frames synthesized by our algorithm has been observed relative to simple bilinear and bicubic interpolation techniques. This improvement is more perceptible in static areas, e.g. the keyboard zone, in which case the algorithm uses almost exclusively the information provided by the spatially accurate high-resolution frames. One can also note that the region in the center of the moving thumb is of better quality inside the frames synthesized by our algorithm, as motion in this area was easy to evaluate because it corresponds to a simple translation of blocks of pixels. However, the detail level near the edge of the thumb inside the frames synthesized by our method is the same as that obtained with the other two techniques. This can be explained by the fact that the algorithm had difficulties calculating the motion of blocks of pixels in these areas, and, therefore, gave less importance to the high-resolution estimate in the reconstruction process.

Also, one may note that the average error per pixel is higher in the images synthesized by our method than in those synthesized by bilinear interpolation. This is likely due to the fact that small frame-to-frame pixel variations resulting from image noise contribute the bulk of the SSD measure. It is obvious, on visual inspection, that our method produces better approximations of the ideal image than the other methods shown; thus, the SSD is not an appropriate metric to quantify the performance of these algorithms.

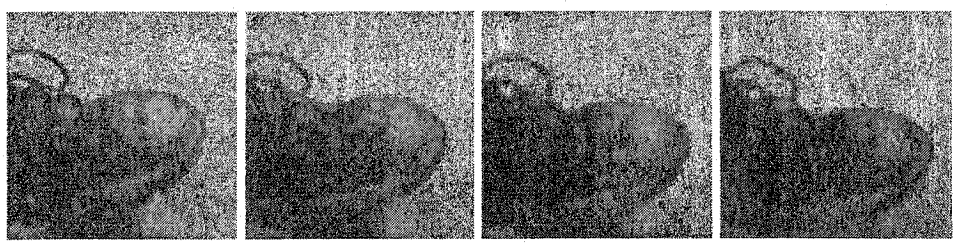
Figure 4.2 shows a case where the motion corresponds to a slight rotation of a boot. Although the motion cannot globally be expressed as a simple translation of pixels, the quality of the frames synthesized by our method is still acceptable. This is due to the small angle of the rotation from frame to frame, which facilitates the tracking of local blocks by the motion evaluation step. However, one may note a slight degradation in the quality of the synthesized frames as a function of time. This can be explained by the fact that the synthesized frames accumulate corrections from frame to frame. In other words, when low-resolution information is introduced into a frame S_t, the next frame S_{t+1} will also suffer from this correction, as each synthesized frame depends on the previous one to obtain high-resolution information, until the acquisition of the next true high-resolution



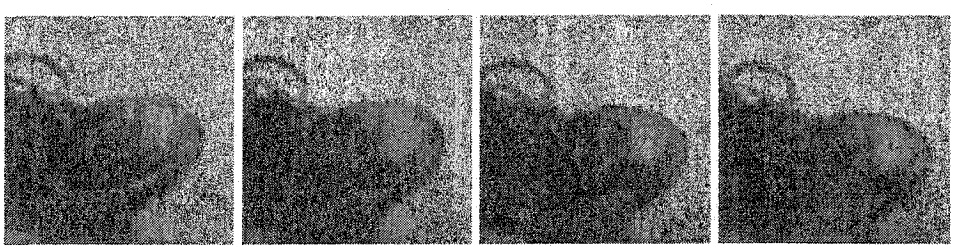
(a) Ideal high-resolution frames



(b) Actual input to algorithm



(c) Frames synthesized by our method



(d) Frames synthesized by bilinear interpolation

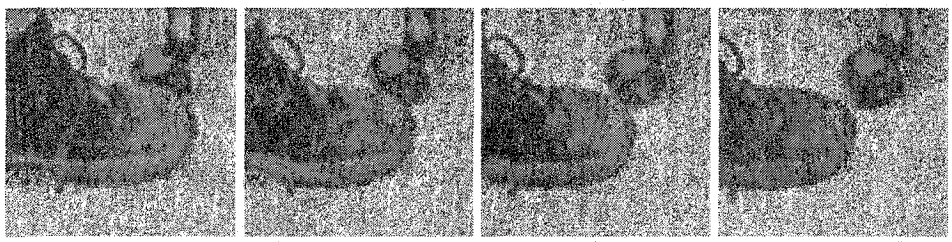
Figure 4.2. Comparison of the simulate-and-correct algorithm with a bilinear technique. In this example, $n = 4$, and the size of the blocks in the motion estimation step is 4×4 low-resolution pixels.

frame.

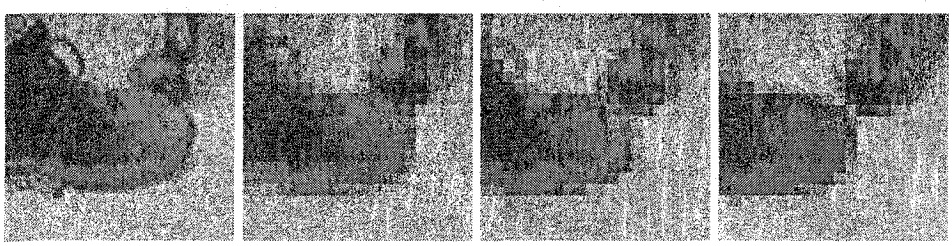
Figure 4.3 illustrates a case for which $n = 16$. That is, during the exposure time of every single high-resolution frame, sixteen low-resolution frames were produced. For space reasons, only the first four frames are shown. As is obvious, the difference in resolution between the high- and low-resolution frames is significant. Also, the difference in quality between the frames synthesized by our algorithm and those generated by bilinear interpolation decreases rapidly. This degradation of the synthesized frames is particularly noticeable on the shoelace. Clearly, a relatively small n should be used for best results.

Figure 4.4 illustrates a pathological case, in which our algorithm cannot follow the motion in the scene. The width of the vertical lines in the ideal frames corresponds to one high-resolution pixel. Thus, assuming $n = 4$, the corresponding low-resolution frames have the same value at each pixel. As a result, it is impossible for the motion evaluation algorithm to detect the translation of the vertical lines and, thus, the second frame synthesized does not exhibit the motion of the corresponding ideal frame.

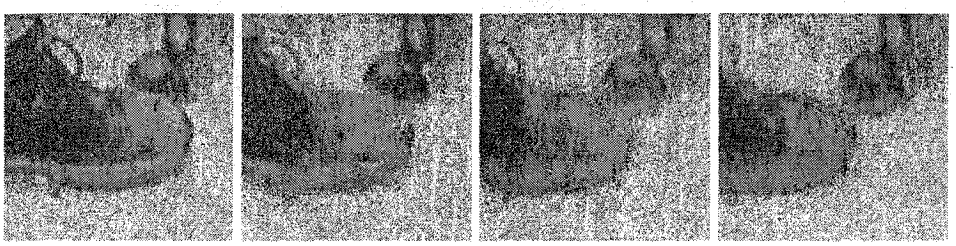
Globally, the previous examples show that the quality of frames generated by our algorithm is always better than or equal to the quality of those generated with a simple interpolation technique. This can be explained by the nature of our algorithm, which uses bilinear interpolation when the estimate step cannot provide better results, and thus, we can guarantee a minimum level of quality. Furthermore, this example shows that scene areas that exhibit a higher level of motion are more difficult to reconstruct and therefore the quality improvement over standard interpolation techniques is insignificant. However, as the human visual system is less sensitive to detail in areas of motion, this factor should not be seen as a serious shortcoming.



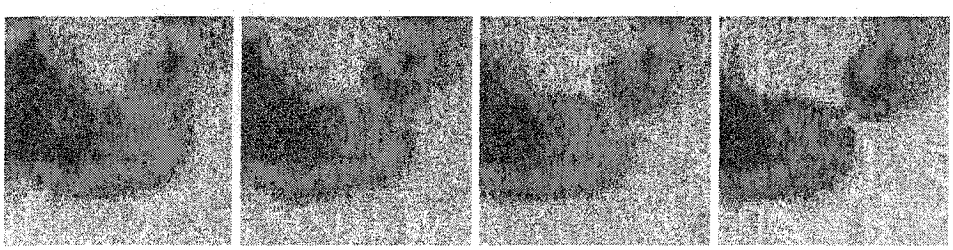
(a) Ideal high-resolution frames



(b) Actual input to algorithm

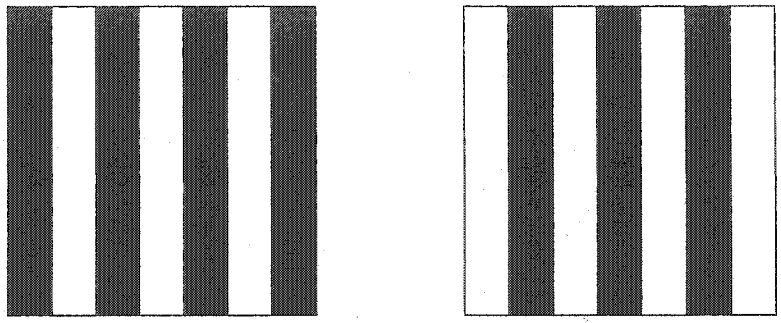


(c) Frames synthesized by our method

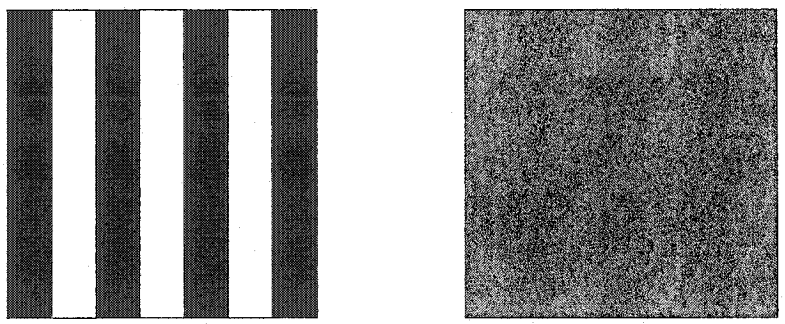


(d) Frames synthesized by bilinear interpolation

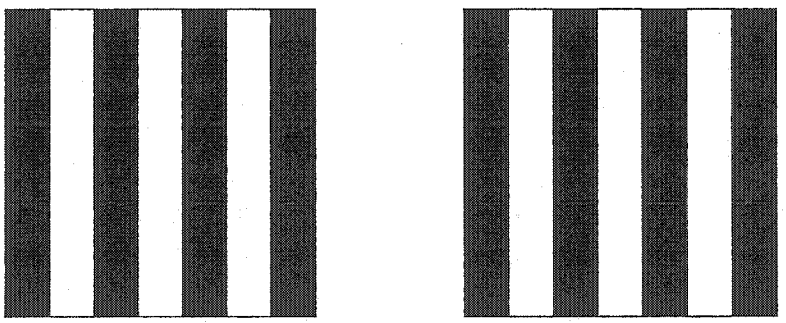
Figure 4.3. Comparison of the simulate-and-correct algorithm with a bilinear technique. In this example, $n = 16$, and the size of the blocks in the motion estimation step is 4×4 low-resolution pixels.



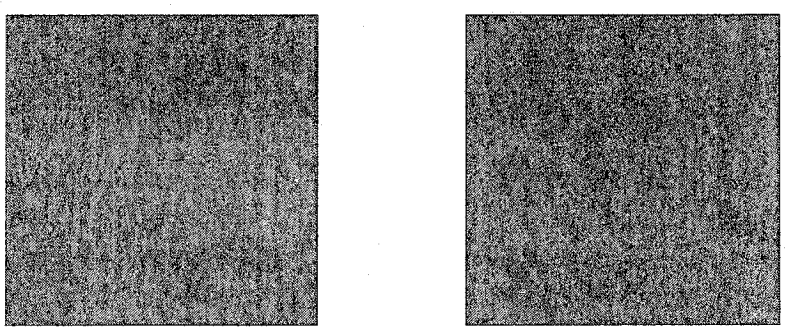
(a) Ideal high-resolution frames



(b) Actual input to algorithm



(c) Frames synthesized by our method



(d) Frames synthesized by bilinear interpolation

Figure 4.4. Results produced by our algorithm when applied to a pathological case, in which it is impossible to detect motion at low-resolution.

Chapter 5

Conclusions and Future Work

A new method of increasing the frame rate of video cameras at high-resolution has been presented. This involves the use of an image processing algorithm in combination with special video hardware that simultaneously produces high-resolution frames at low frequency and low-resolution frames at high frequency. The method combines the spatially optimal high-resolution information with the temporally optimal low-resolution information to approximate an ideal high-resolution representation of the scene.

Our high-resolution video synthesis technique demonstrates a significant improvement in quality relative to simple bilinear or bicubic interpolation techniques. While the frames produced by our algorithm are always of equal or greater quality than those produced by simple interpolation techniques, the qualitative improvement tends to be more perceptible in the areas of the scene involving little or no motion. Nonetheless, the algorithm still performs well when higher velocity, translation-based motion occurs. Areas located at the boundary of two zones presenting different motion characteristics are more difficult to reconstruct, as these tend to change significantly from one frame to the next, e.g., due to occlusion or disocclusion, thereby posing a challenge to the matching task of the motion evaluation step. In such cases, the pixels generated by our algorithm are produced by a simple interpolation technique, which does not significantly affect the quality of the resulting frames, as human vision is less sensitive to detail in areas of motion. Furthermore, the simplicity of the algorithm facilitates its hardware implementation. While the bottleneck

of the presented method is the motion evaluation step, the size of the search window can be adapted to satisfy time constraints.

Future work will include the enhancement of the motion estimation step in the estimate synthesis, which currently uses a block matching algorithm to evaluate the motion. Since the motion model assumes that all pixels in a given block move together, the motion evaluation is necessarily coarse. Although this technique is efficient, it yields reduced accuracy zone boundaries exhibiting different motion characteristics. This could be improved by refining the process inside those blocks overlapping a moving object and the background. Such blocks could be identified easily by using the value returned by equation 3.1. The technique could be improved further by making use of the information contained in the underexposed high-resolution frame, produced by the camera. While we already use this information to reduce motion blur in high-resolution frames, it could also be exploited for resolution enhancement, in particular in regions of high luminosity, as these provide additional information that is currently under-utilized. Finally, another method for unblurring the high-resolution frames could be explored. Since motion information is contained in the low-resolution frames, one could use these, instead of the underexposed high-resolution frames, to reduce motion blur in the synthesized frames. While we expect this to yield somewhat inferior results, this method offers a compelling advantage of not requiring complex video hardware, as no intermediate high-resolution frames are needed.

Bibliography

[1] G. Holst, *CCD Arrays, Cameras, and Displays*. Winter Park, FL: JCD Publishing, 1996.

[2] R. C. Gonzalez and P. Wintz, *Digital Image Processing*. New York: Addison-Wesley, 1987.

[3] W. K. Pratt, *Digital Image Processing*. New York: Wiley, 1991.

[4] A. K. Jain, *Fundamentals of Digital Image Processing*. Eaglewood Cliffs, NJ: Prentice-Hall, 1989.

[5] R. L. Lagendijk and J. Biemond, *Iterative Identification and Restoration of Images*. Boston, MA: Kluwer, 1991.

[6] D. C. Youla, "Generalized image restoration by the method of alternating orthogonal projections," *IEEE Trans. Circuits Syst.*, vol. CAS-25, pp. 694–702, 1978.

[7] J. Hadamard, *Lectures on the Cauchy Problem in Linear Partial Differential Equations*. Yale University Press, New Haven, CT, 1923.

[8] P. R. Smith, "Bilinear interpolation of digital images," *Ultramicroscopy*, vol. 6, pp. 201–204, 1981.

[9] R. Carlson and E. Fritsch, "Monotone piecewise bicubic interpolation," *SIAMJ. Numer. Anal.*, vol. 22, pp. 386–400, Apr. 1985.

[10] R. Y. Tsai and T. S. Huang, "Multiframe image restoration and registration," *Advances in Computer Vision and Image Processing*, vol. 1, pp. 317–339, 1984.

[11] R. A. Roberts and C. T. Mullis, *Digital Signal Processing*. New York: Addison-Wesley, 1987.

[12] S. P. Kim, N. K. Bose, and H. M. Valenzuela, "Recursive reconstruction of high resolution image from noisy undersampled multiframes," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, pp. 1013–1027, June 1990.

[13] R. R. Schultz and R. L. Stevenson, "Extraction of high-resolution frames from video sequences," *IEEE Trans. Image Processing*, vol. 5, no. 6, pp. 996–1011, June 1996.

[14] B. C. Tom, A. K. Katsaggelos, and N. P. Galatsanos, "Reconstruction of a high-resolution image by simultaneous registration, restoration and interpolation of low-resolution images," in *Proc. IEEE International Conference on Image Processing*, vol. 2, Washington, DC, 1995, pp. 539–542.

[15] S. Borman and R. Stevenson, "Spatial resolution enhancement of low-resolution images sequences - a comprehensive review with directions for future research," July 1998. [Online]. Available: citeseer.nj.nec.com/borman98spatial.html

[16] M. Elad and A. Feuer, "Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images," *IEEE Trans. Image Processing*, vol. 6, no. 12, pp. 1646–1658, Dec. 1997.

[17] D. Keren, S. Peleg, and R. Brada, "Image sequence enhancement using subpixel displacements," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 1988, pp. 742–746.

[18] M. Irani and S. Peleg, "Motion analysis for image enhancement: Resolution, occlusion and transparency," *Journal of Visual Communications and Image Representation*, vol. 4, pp. 324–335, Dec. 1993.

[19] S. Haykin, *Adaptive Filter Theory*. Prentice-Hall, 1986.

[20] C. K. Chui and G. Chen, *Kalman Filtering*. New York: Springer-Verlag, 1990.

[21] A. Zomet and S. Peleg, "Multi-sensor super-resolution," in *IEEE Worshop on Applications of Computer Vision*, Orlando, Dec. 2002, pp. 27–31.

[22] E. L. Turner, W. A. Hill, and D. L. Wilson, "High speed digital camera," U.S. Patent US 6 198 505 B1, Mar. 6, 2001.

[23] ISO/IEC 11172-2, "Information technology - coding of moving pictures and associated audio for digital storage media at up to about 1.5 mbits/s - part 2: video," 1993.

[24] ISO/IEC 13818-2 and ITU-T Recommendation H.262, "Information technology - generic coding of moving pictures and associated audio information: Video," 1995.

[25] ITU-T Recommendation H.261, "Video codec for audiovisual services at p $\times$ 64 kbits/s," Mar. 1993.

[26] ITU-T Recommendation H.263, "Video coding for low bit rate communication," Feb. 1998.

[27] T. Komarek and P. Pirsch, "Array architecture for block matching algorithms," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 1301–1308, Oct. 1989.

[28] D. Vos and M. Stegherr, "Parametrizable vlsi architectures for full-search block-matching algorithms," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 1309–1316, Oct. 1989.

[29] K. Yang, M. Sun, and L. Wu, "A family of vlsi designs for the motion compensation block-matchnig algorithm," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 1317–1325, Oct. 1989.

[30] S. Kemeny and Al., "Multiresolution image sensor," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 575–583, Aug. 1997.

[31] Z. Zhou, B. Pain, and E. Fossum, "A cmos imager with on-chip variable resolution for light-adaptative imaging," in *Proc. IEEE Int. Solid-State Circuits Conf.*, San-Francisco, Feb. 1998, pp. 174–175.

[32] X. Liu and A. E. Gamal, "Simultaneous image formation and motion blur reduction via multiple capture," in *Proc. International Conference on Acoustic, Speech and Signal Processing*, Salt Lake City, May 2001.

[33] D. Yang, A. E. Gamal, B. Fowler, and H. Tian, "A 650x512 cmos image sensor with ultra-wide dynamic range floating-point pixel level adc," *IEEE J. Solid-State Circuits*, vol. 34, no. 12, pp. 1821–1834, Dec. 1999.

[34] R. Polana and R. C. Nelson, "Recognition of nonrigid motion," in *Proceedings DARPA Image Understanding Workshop*, Monterey, CA, Nov. 1994, pp. 1219–1224.

[35] Y.-S. Chen, Y.-P. Hung, and C.-S. Fuh, "Fast block matching algorithm based on the winner-update strategy," *IEEE Trans. Image Processing*, vol. 10, no. 8, pp. 1212–1222, Aug. 2001.