High-Resolution Video Synthesis from Mixed-Resolution Video Based on the Estimate-and-Correct Method

Stéphane Pelletier, Stephen P. Spackman and Jeremy R. Cooperstock
Department of Electrical and Computer Engineering
McGill University, Montréal, Canada
stephane|stephen|jer@cim.mcgill.ca

Abstract

A technique for increasing the frame rate of CMOS video cameras is presented. The method uses the non-destructive readout capabilities of CMOS imagers to obtain low-speed, high-resolution frames and high-speed, low-resolution frames simultaneously. The algorithm translates the pixels of the full resolution images with respect to the motion dynamics observed in the low-resolution frames and corrects the result as necessary for consistency with the low-resolution frames. Noting that due to the longer exposure time required, high-resolution frames are more prone to motion blur than low-resolution frames, and thus, 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.

1. Introduction

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 [4]. 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 increase the frame rate at high resolution of CMOS image sensors, we propose using their non destructive readout capabilities to simultaneously generate high-resolution frames H at frame rate h and low-resolution frames L at frame rate l, as depicted in Figure 1. 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, the frame rate l of the low-resolution sequence is naturally higher than h. The high- and low-resolution frames represent the same scene and are used respectively to capture high-frequency details and object motion. Our method applies an image-processing algorithm to both sequences of frames H and L in order to synthesize a high-resolution video sequence

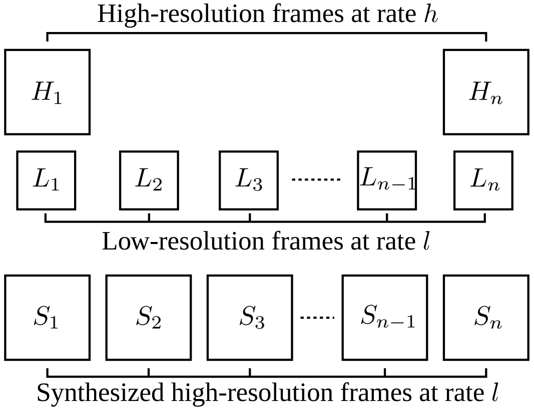

High-resolution frames at rate h
H_1
H_n
L_1
L_2
L_3
L_{n-1}
L_n
Low-resolution frames at rate l
S_1
S_2
S_3
S_{n-1}
S_n
Synthesized high-resolution frames at rate l

Figure 1. Synthesis of high-resolution video at frame rate l from low-resolution video at frame rate l and high-resolution video at frame rate h.

S, at high frame rate l, containing the detail of the high-resolution frames H and the motion dynamics of the low-resolution frames L.

A motion evaluation algorithm is used to evaluate pixel motion in a coarse manner 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 containing mostly high-resolution and some low-resolution features. The latter, which are smoothed by simple interpolation, tend to appear when abrupt motion occurs in the scene. Because high-resolution frames require a longer exposure time, these are more sensitive to motion blur and thus, the algorithm includes a step to reduce blur to a level equivalent to that of the low-resolution frames.

2. Previous Work

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. A review of such super-resolution techniques is presented by Borman *et al* [1]. These can be divided into frequency domain [13][9][6] and spatial domain methods [10][12][3]; the former tend to be simpler and are preferred for applications involving global translation motion, which may happens for example when the camera moves laterally. Spatial domain methods, however, are better suited for general video sequences that may contain local motion.

Elad and Feuer [3] proposed a technique based on adaptive filtering theory, using a time and space filter that operates on a set of low-resolution images. The restoration procedure solves a large set of sparse linear equations to produce a sequence of images at higher resolution. This technique depends on prior evaluation of motion between low-resolution frames. Shechtman *et al* [11] extended the notion of super-resolution to the space-time domain. Their method combines several video sequences of different resolutions and frame rates in order to produce a single video sequence of better quality. A trade-off between spatial and temporal resolution is achieved in the sense that increasing one is done at the expense of the other. A particular case of their method consists of using a low-resolution video sequence with two high-resolution images in order to produce a high-resolution video sequence. This application is closely related to that presented in this paper.

3. Frame Acquisition Model

A camera produces a video sequence by capturing images of a scene at regular time intervals, which define the frame rate. 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 corresponds to 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 $Q(t)$ read out at the end of the exposure time T. The amount of charge accumulated in a photosite is a linear function of the incident illumination intensity and the integration period.

The light sensitivity of a photosite depends on the size of its reception surface; a smaller surface is less sensitive and thus, requires a greater minimal exposure time, T_{min}, to produce a pixel at a certain light intensity. If the charge accumulated at a photosite is read after a shorter exposure time, for example, $t = \frac{T_{min}}{4}$, the value will not be reliable. However, if the capacitor values for adjacent photosites are added by groups of four, thus corresponding to virtual photosites whose reception surface is four times larger, the summed values would be sufficient to produce a reliable low-resolution frame. This technique is, in fact, used in some multiresolution video devices [5][16], which can be programmed to generate frames at different resolutions. These devices can increase their light sensitivity at the cost of resolution. Therefore, by adding the values of adjacent pixels, it is possible to generate a low-resolution frame every $\frac{T_{min}}{n}$ and a high-resolution frame every T_{min} time period, where n gives the number of high-resolution pixels that are combined to produce a low-resolution one. 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 frame rate provided by the camera.

An object moving within the scene during the exposure period 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 time required for the acquisition of a high-resolution frame is approximately four times that for a low-resolution one, the captured high-resolution frames are more prone to motion blur, as illustrated in Figure 2. The first row shows four low-resolution frames that were captured successively by the video camera, using an exposure time of $\frac{T_{min}}{4}$ for each, while the second row shows the acquisition process of the equivalent high-resolution frame at different stages, each corresponding to the end of the exposure of the low-resolution frame above it. By the time we complete the ex-

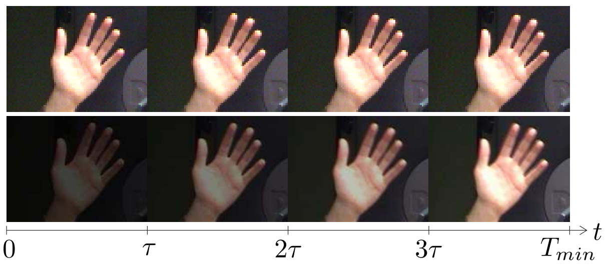
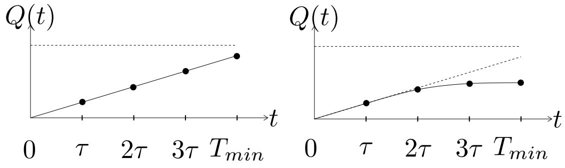


Figure 2. Sensitivity to motion blur as a function of the exposure time.

posure of the high-resolution frame at $t = T_{min}$, this image has accumulated four times the amount of blur of one low-resolution frame. Since our technique translates pixels of captured high-resolution frames in order to synthesize successive frames, sensitivity to motion blur in the synthesized frames will also correspond to an exposure time of T_{min}, rather than $\frac{T_{min}}{4}$. Ideally, we would like to avoid these problems of motion blur.

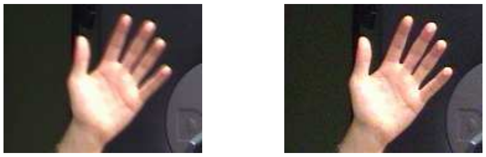
Liu *et al* [8] present a method that uses special video hardware capable of capturing many images within a normal exposure time and takes advantage of the extra information provided by these underexposed images to help reduce the level of motion blur in the generated frames. Figure 3(a) represents the charge accumulation 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 slope, i.e. the photocurrent, is constant. On the other hand, Figure 3(b) indicates the effect of motion on photocurrent. Under such conditions, an accurate measure of photocurrent at the beginning of the exposure time cannot be obtained from $Q(T_{min})$. Instead, Liu uses the different values of $Q(t)$ measured during an exposure to estimate the photocurrent at the beginning of the exposure time $(t = 0)$. His method determines whether motion has occurred during the exposure by examining the general shape of the integration curve. If motion is detected, the point t_m corresponding to the beginning of motion is identified. For example, in the case of Figure 3(b), $t_m = 2\tau$. Then, only the information from $t = 0$ to $t = t_m$ is used to estimate the photocurrent at the beginning of the exposure. Otherwise, all the points are used to produce the estimate, in which case, the accuracy is improved.



(a) (b)

Figure 3. Effect of scene motion on the photocurrent. (a) no motion. (b) motion.

For the purpose of reducing blur in the high-resolution frames produced by our special video device, 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 high-resolution frame we attempt to reconstruct. Using a similar approach to Liu *et al*, we can estimate this value and thus, reduce motion blur in the generated high-resolution frames, as shown in Figure 4.



(a) (b)

Figure 4. Motion blur reduction in a high-resolution frame. (a) Frame affected by motion blur (b) Frame after motion blur reduction.

4. Estimate-and-Correct Method

The strategy used to synthesize the high-resolution frame S_t involves translating the pixels of the previous frame S_{t-1} with respect to motion observed at low resolution. The problem with this approach is that motion cues alone are insufficient to describe the scene changes between the two frames. 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. Clearly, the quality of motion evaluation depends on the nature of scene changes; when object motion is rapid or 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 the necessary motion estimation generally increases with the complexity of 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. This approach 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 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.

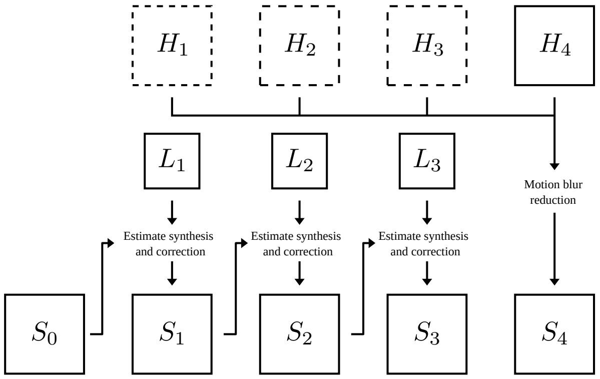

H_1
H_2
H_3
H_4
L_1
L_2
L_3
Motion blur reduction
Estimate synthesis and correction
Estimate synthesis and correction
Estimate synthesis and correction
S_0
S_1
S_2
S_3
S_4

Figure 5. Steps involved in the algorithm.

Figure 5 illustrates the application of our technique. When the integration period of a high-resolution frame is completed, this frame and its three underexposed predecessors (dashed) are used to synthesize a high-resolution frame with reduced motion blur in it. In other cases, the estimate-and-correct method is applied between the current low-resolution frame and the last synthesized frame.

A widely used method of evaluating motion between two frames in a video sequence is the block matching algorithm (BMA). Because of the wide distribution of MPEG video encoding applications, this step can be performed by readily available motion estimation hardware [7][14][15]. Based on the assumption that each area of the current frame F_t can be obtained from the translation of some corresponding area in the previous frame F_{t-1}, the BMA partitions F_t into equal-sized non-overlapping square blocks and finds, for each, the best matching block in F_{t-1}. The displacements of these best-matched blocks are represented as vectors, describing how the different parts of the scene moved between the two frames. Because the motion representation of all the pixels in a block is reduced to a single motion vector, the evaluation computed by the BMA is necessarily coarse. This may be problematic in some situations; for example, when a block contains both the edge of a moving object and a portion of a static background, block motion cannot correctly translate the object pixels without affecting the background as well.

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}:

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

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

$$(u_b, v_b) = \arg\min \mathrm{SSD}_{(x,y)}(u,v). \qquad (2)$$

The BMA is the most critical component of the high-resolution video synthesis algorithm. Its efficiency is key to reducing execution time [2] and its accuracy determines the quality of the synthesized high-resolution frames. While a 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. One way to accelerate the block-matching process is to restrict the search area to a small neighbourhood; thus the number of potential candidates to examine 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.

Figure 6 shows an example of the estimation step, in which a foot is moving backward (toward the left) in front of the wheel of a chair. Figure 6(a) corresponds to the high-resolution frame that was synthesized at time $t - 1$, Figure 6(b) is the current low-resolution frame L_t, and Figure 6(c) corresponds to the estimate E_t, obtained by applying to S_{t-1} the motion dynamic observed between S_{t-1} and L_t. The estimate E_t will typically contain artifacts similar to those produced by a poor MPEG codec, mainly due

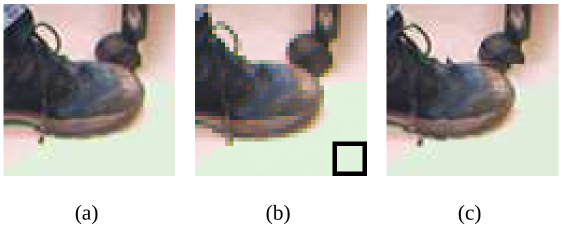

(a) (b) (c)

Figure 6. Estimate synthesis by translation of blocks (size depicted by a square). (a) S_{t-1}. (b) L_t. (c) E_t.

to the fact that it is not always possible to £nd an adequate match in S_{t-1} for each pattern block in L_t. For example, observing the estimate in Figure 6, 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.

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.

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:

$$M(x,y) = \frac{[L_t(x,y) - E_t^{'}(x,y)]^2}{K^2} \tag{3}$$

where K is a chosen constant, based on the maximum possible color component value, to scale the values of $M(x,y)$ to $[0,1]^1$. 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. The merging process can be expressed by the following equation

$$\begin{aligned} S_t(x,y) = &\big(1 - M(x,y)\big)E_t(x,y) \\ &+ M(x,y)B_t(x,y) \end{aligned} \tag{4}$$

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.

5. Experimental Results

To validate our approach to high-resolution video synthesis, a qualitative comparison was performed between bilinear interpolation and our proposed method. Since the mixed-resolution video hardware does not exist yet, its output had to be simulated. A sequence of ideal high-resolution frames was captured using a conventional £xed resolution camera. Low-resolution frames, as well as underexposed, blurred high-resolution frames were then generated from these ideal frames. 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 7 shows a comparison between bilinear interpolation and our technique using the same example as in Figure 6. For space reasons, results are presented for one frame of the sequence, which represents a close-up of a foot moving in front of a chair. Figure 7(a) shows the ideal high-resolution frame, namely the original frame from which the low-resolution frame in Figure 7(b) was simulated. This frame is not available for the algorithm and is presented here for comparison purposes. Figure 7(c) shoes the frame synthesized by our algorithm, which is actually the corrected version of Figure

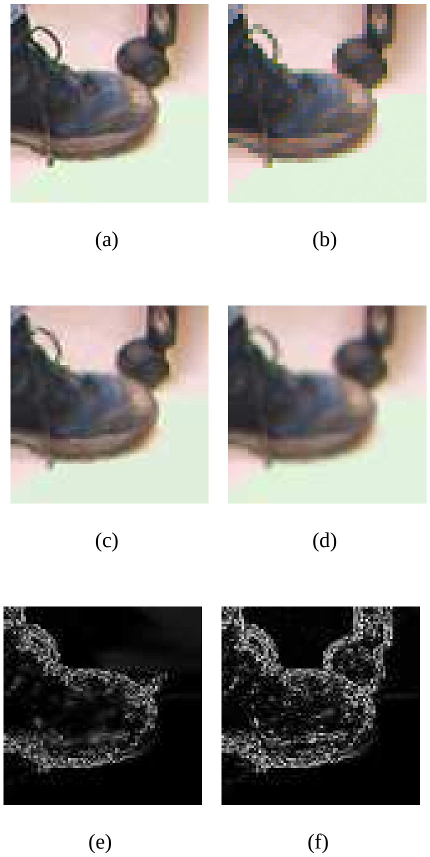


Figure 7. Comparison of the estimate-and-correct algorithm with bilinear interpolation.

[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.

6(c). Figure 7(d) presents the corresponding result obtained using bilinear interpolation. Figure 7(e) and Figure 7(f) respectively show the error in the frame synthesized by our method and that in the frame produced using bilinear interpolation. These error maps were produced by taking the difference between the synthesized frames and the ideal frame of Figure 7(a).

In this example, a significant improvement in quality of the frame synthesized by our algorithm has been observed relative to simple bilinear interpolation. This improvement is more perceptible in static areas, e.g. the wheel, in which case the algorithm uses, almost exclusively, the information provided by the spatially accurate high-resolution frame. One can also note that the region in the center of the moving foot 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 foot inside the frame synthesized by our method is similar to that obtained by the other technique. 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.

This example demonstrates that the quality of frames generated by our algorithm is always greater than or equal to the quality of those generated by simpler interpolation techniques. This can be explained by the nature of our algorithm, which falls back to bilinear interpolation when the estimate step cannot offer improved results. Furthermore, scene areas that exhibit a higher level of motion are more difficult to reconstruct and therefore the quality improvement of our algorithm 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 considered as a serious shortcoming.

6. Conclusions and Future Work

A new method for increasing the frame rate of video cameras at high-resolution has been presented. The method combines spatially optimal high-resolution information with temporally optimal low-resolution information to approximate an ideal high-resolution representation of the scene. 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 includes the enhancement of the motion estimation step in the estimate synthesis. Since the block matching motion model assumes that all pixels in a given block move together, the evaluation is necessarily coarse. Although this technique is efficient, it yields reduced accuracy at zone boundaries exhibiting different motion characteristics. This could be improved by refining the process inside those blocks overlapping a moving object and the background.

References

[1] S. Borman and R. Stevenson. Spatial resolution enhancement of low-resolution images sequences - a comprehensive review with directions for future research, July 1998.

[2] Y.-S. Chen, Y.-P. Hung, and C.-S. Fuh. Fast block matching algorithm based on the winner-update strategy. 10(8):1212–1222, Aug. 2001.

[3] M. Elad and A. Feuer. Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images. 6(12):1646–1658, Dec. 1997.

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

[5] S. Kemeny and Al. Multiresolution image sensor. 7:575–583, Aug. 1997.

[6] S. P. Kim, N. K. Bose, and H. M. Valenzuela. Recursive reconstruction of high resolution image from noisy undersampled multiframes. 38:1013–1027, June 1990.

[7] T. Komarek and P. Pirsch. Array architecture for block matching algorithms. 36:1301–1308, Oct. 1989.

[8] 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.

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

[10] R. R. Schultz and R. L. Stevenson. Extraction of high-resolution frames from video sequences. 5(6):996–1011, June 1996.

[11] E. Shechtman, Y. Caspi, and M. Irani. Increasing space-time resolution in video. In *Proceedings of the 7th European Conference on Computer Vision-Part I*, pages 753–768, May 2002.

[12] 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*, volume 2, pages 539–542, Washington, DC, 1995.

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

[14] D. Vos and M. Stegherr. Parametrizable vlsi architectures for full-search block-matching algorithms. 36:1309–1316, Oct. 1989.

[15] K. Yang, M. Sun, and L. Wu. A family of vlsi designs for the motion compensation block-matchnig algorithm. 36:1317–1325, Oct. 1989.

[16] 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.*, pages 174–175, San-Francisco, Feb. 1998.