Published in IET Computers & Digital Techniques
Received on 10th June 2009
Revised on 3rd February 2010
doi: 10.1049/iet-cdt.2009.0053

ISSN 1751-8601

Blur identification with assumption validation for sensor-based video reconstruction and its implementation on field programmable gate array

M.E. Angelopoulou C.-S. Bouganis P.Y.K. Cheung

Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2BT, UK
E-mail: m.angelopoulou@imperial.ac.uk

Abstract: Restoration methods, such as super-resolution (SR), largely depend on the accuracy of the point spread function (PSF). PSF estimation is an ill-posed problem, and a linear and uniform motion is often assumed. In real-life systems, this may deviate significantly from the actual motion, impairing subsequent restoration. To address the above, this work proposes a dynamically configurable imaging system that combines algorithmic video enhancement, field programmable gate array (FPGA)-based video processing and adaptive image sensor technology. Specifically, a joint blur identification and validation (BIV) scheme is proposed, which validates the initial linear and uniform motion assumption. For the cases that significantly deviate from that assumption, the real-time reconfiguration property of an adaptive image sensor is utilised, and the sensor is locally reconfigured to larger pixels that produce higher frame-rate samples with reduced blur. Results demonstrate that once the sensor reconfiguration gives rise to a valid motion assumption, highly accurate PSFs are estimated, resulting in improved SR reconstruction quality. To enable real-time reconstruction, an FPGA-based BIV architecture is proposed. The system's throughput is significantly higher than 25 fps, for frame sizes up to 1024×1024, and its performance is robust to noise for signal-to-noise ratio (SNR) as low as 20 dB.

1 Introduction

The work presented in this paper lies in the area of real-time video processing and focuses on the problem of enhancing in real time the spatio-temporal resolution of the captured video sequence [1]. To achieve the above, this work explores, proposes and brings together into a novel imaging system appropriate imaging techniques related to different levels of processing: a high-level video enhancement algorithm, low-level implementation on reconfigurable hardware, field programmable gate arrays (FPGAs) in particular and state-of-the-art image sensor technology [2, 3]. Contrary to traditional cameras, which are passive, the proposed imaging system is dynamically configured in real time according to the captured video data, based on the real-time interaction of an adaptive image sensor with an FPGA-based processing unit. The FPGA both configures the adaptive image sensor in a manner that maximises the captured information and further processes these raw data to render outputs of high resolution both in space and in time. Therefore, this work proposes a sensor configuration scheme and an appropriate processing method for video enhancement under real-time constraints. This method is based on a blur identification and validation technique that improves the reconstruction quality of the final output, and a hardware architecture of the method is proposed and implemented on FPGA. The throughput that is achieved is significantly higher than the 25 fps real-time requirement, for frame sizes up to 1024×1024, and the system's performance is robust to noise for signal-to-noise ratio (SNR) as low as 20 dB.

Blur identification is a critical task when motion blur degrades the quality of the captured video sequence. The ill-posed nature of the identification problem and the required high computational cost pose a bottleneck in the overall system performance, in particular when real-time applications are targeted. In the literature, motion blur identification is simplified by employing a linear and uniform assumption for the motion point spread function (PSF) [4–9], where 'linear' implies a motion trajectory that is accurately approximated with a first-order polynomial and 'uniform' indicates identical PSF weights. This reduces the identification task to the estimation of two parameters, namely the direction and the extent of the underlying motion, thus resolving the ill-posed nature of the problem and decreasing the related computational load. However, in real-life systems, the actual blurring function might deviate significantly from an ideal linear and uniform PSF [1], producing inaccurate blur estimates that directly affect subsequent restoration.

In [2], a video enhancement system based on an adaptive image sensor [3, 10] is proposed. Possible sensor configurations that maximise the captured raw information are explored and are combined with processing methods that increase the spatio-temporal resolution of the output. Two approaches are presented, a deconvolution-based and a super-resolution (SR)-based approach. For highly dynamic regions, the SR-based approach is shown to be more appropriate. In this approach, each motion region is locally configured to a uniform grid of large pixels, rendering high frame-rate samples with reduced motion blur, whereas areas with slow motion or no motion are configured to the elementary pixels of the sensor. The spatial resolution of the motion areas is increased by fusing the high frame-rate samples with SR techniques. However, for very fast motion, these samples are inevitably blurred themselves, and the fidelity of the SR output is bounded by the blur effect.

This work addresses the above shortcomings by proposing an interaction of a joint blur identification and validation (BIV) scheme with an adaptive image sensor. Contrary to existing blur identification methods, which estimate linear motion parameters without validating the linearity and uniformity assumption [4–9], the proposed BIV block also identifies cases that significantly deviate from the initial assumption. This keeps the blur identification process simple, since linear and uniform motion is assumed, while identifying cases where the assumption is invalid and thus the PSF estimation is inaccurate. Such cases are resolved by configuring the adaptive sensor to larger pixels that produce samples with reduced blur. Once the appropriate pixel size is employed for which the linear motion assumption is valid, accurate PSFs are estimated, increasing the SR reconstruction quality.

The BIV block employs the autocorrelation framework for blur identification [4], which is extended so as to perform assumption validation in addition to blur identification. To target real-time restoration, the BIV block is implemented on reconfigurable hardware, an FPGA in particular. By exploiting the parallelism, pipelining and data reuse possibilities offered by an FPGA, high throughput is achieved, which meets the strict performance constraints of real-time applications.

In summary, the major contribution of this work is the proposal of a unifying approach that brings together imaging techniques related to different levels of processing: algorithmic video enhancement, FPGA-based video processing and adaptive image sensor technology. Contrary to the traditional passive cameras, the proposed imaging system is dynamically configurable and thus able to adapt optimally to the given video data. More specific contributions of this work are as follows:

1. A method that performs BIV is proposed. The method utilises the autocorrelation-based blur identification framework of [4], which is extended so as to include assumption validation.
2. A methodology is proposed that includes the interaction of an adaptive image sensor with the BIV scheme. The proposed methodology increases the accuracy in the estimation of the PSFs related to the SR inputs, thus improving the SR reconstruction quality.
3. A hardware architecture of the BIV scheme is presented. The proposed architecture is implemented on an FPGA and its performance is evaluated.

It should be noted that the proposed joint BIV scheme could also be used for general-purpose blur identification and can be combined with different restoration methods [11–13]. To the best of our knowledge, this is the first hardware approach to the blur identification problem that is reported in the literature. As demonstrated in the paper, taking advantage of parallelism and data reuse, the FPGA-based implementation of the BIV scheme renders a throughput significantly higher than the 25 fps real-time requirement, for frame sizes up to 1024×1024, while giving robustness to noise for SNR as low as 20 dB. Moreover, individual high-throughput blocks that have been proposed and implemented as part of the BIV architecture, such as the one implementing the autocorrelation function (ACF), are useful for a variety of image and signal processing applications.

The remainder of the paper is organised as follows. Section 2 presents the detailed algorithm of the proposed joint BIV scheme. Section 3 provides the motivation for developing the above scheme and explains how the scheme is effectively incorporated into a video enhancement system that is based on an adaptive image sensor. The SR reconstruction quality of the resulting system is evaluated in Section 4, which provides software results for different PSF assumptions and various motion types. These results demonstrate that the interaction of the joint BIV scheme with the reconfiguration property of an adaptive sensor significantly improves the reconstruction quality. The hardware architecture of the proposed BIV scheme, which achieves real-time performance, is described in Section 5. Section 6 presents results from the hardware implementation. Specifically, Section 6.1 discusses the throughput and hardware requirements of the proposed architecture for different sets of parameters, and Section 6.2 evaluates the system performance for various parameters and different levels of noise. Finally, Section 7 concludes the paper.

2 Joint identification and validation: description of the algorithm

The proposed joint BIV scheme utilises the spatial domain blur identification framework of [4], which is extended in order to incorporate as well validation of the initial motion assumption. The decision of using a spatial domain instead of a frequency domain method is based on both algorithmic and hardware-related criteria. Specifically, frequency domain blur identification methods [7–9] are restricted to blurring functions that exhibit a periodic pattern of spectral zeros, which is not always the case [4]. Moreover, a spatial method reduces the required hardware cost by avoiding the transformations between the two domains and also removing the need to handle outputs with large dynamic range, as produced by frequency domain analysis. Therefore, the need for long word-lengths in a hardware implementation is not required. Among the spatial blur identification methods [4–6], the autocorrelation-based method of [4] was selected because of its potential for maximum parallelism and data reuse, as it will be demonstrated in Section 5.

The detailed BIV algorithm is next presented. The following notation has been used:

c	classification flag, 1/0: valid/invalid linearity and uniformity assumption
θ, L	estimated direction and extent of motion (in pixels)
f	input image
Δ_ϕ	directional image derivative operator in direction ϕ
$I(\phi)$	normalised total intensity (TI) of $\Delta_\phi(f)$
d	step in degrees for calculating $\Delta_\phi(f)$ and $I(\phi)$
g	horizontal derivative of the derivative vertical to θ
M	number of rows of g rotated by θ

doi: 10.1049/iet-cdt.2009.0053

K	number of columns of g rotated by θ minus 1
$m_i[n]$	the ith line of g that derives after interpolation along θ, where $m_i[n] = 0$ for $n \notin [0, K]$
R_i	discrete set of autocorrelation coefficients for $m_i[n]$
$\bar{R}$	mean of the discrete autocorrelation coefficients of the M lines of g along θ
μ	mask for isolating the moving object from the background
p_1	threshold for identifying considerably small values of L that are owing to motion non-linearities
P, N	number of dominant positive and negative lobes in $\bar{R}$
p_2	threshold for $I(\theta)$

The detailed algorithm, which implements the proposed joint BIV scheme, is given in Fig. 1.

Inputs: f, μ, d, p_1, p_2

Outputs: c, θ, L

1: // ——————— blur identification: ——————— //
2: **for** $\phi = 0^o : d : 180^o - d$ **do**
3: calculate $\Delta_\phi(f)$ and $I(\phi)$
4: find $\theta \in [0^o, 180^o)$: $I(\theta) = \min(I)$
5: $g = \Delta_\theta(\Delta_{\theta+90^o}(f))$
6: $g = g \cdot \mu$
7: **for** $i = 1 : M$ **do**
8: **for** $k = -K : K$ **do**
9: $R_i[k] = \sum_{n=-K}^{K} m_i[n] m_i[n-k]$
10: $\bar{R} = (\sum_{j=1}^{M} R_j)/M$
11: find $L \in [0, K]$: $\bar{R}[L] = min(\bar{R})$
12: // ——————— classification: ——————— //
13: **if** $L < p_1$ **then**
14: $c \leftarrow 0$; **return** c
15: **else**
16: count positive(P) and negative(N) dominant lobes in $\bar{R}$
17: **if** $P \neq 1$ or $N \neq 2$ **then**
18: $c \leftarrow 0$; **return** c
19: **else if** $I(\theta) > p_2$ **then**
20: $c \leftarrow 0$; **return** c
21: **else**
22: $c \leftarrow 1$; **return** c, θ, L

Fig. 1 *Algorithm implementing the proposed joint BIV scheme*

2.1 Blur identification

The blur identification part in the BIV algorithm employs the framework of [4], which is based on the calculation of the mean ACF along the direction of the motion. This subsection discusses the approach of [4], and how this was modified and extended to achieve local, computationally efficient and robust to noise blur identification.

In the lines 2–11 of the BIV algorithm, the linear motion parameters (θ, L) are identified employing the autocorrelation-based framework of [4]. The method of [4] is based on the fact that along the motion direction image smoothness is higher and pixels are correlated. As demonstrated in the BIV algorithm, a series of filtering operations precedes the autocorrelation calculations. First, the image derivatives are calculated in different directions and the total intensities of the derivatives are computed (lines 2 and 3). The direction with the minimum intensity is the motion direction θ. An image derivative g is then generated as follows. First, the derivative of the input frame is calculated in the direction vertical to θ, which removes object-related image properties and emphasises motion-related properties. On this output, the derivative in direction θ is then applied (line 5). Image g is then traversed along direction θ, and the autocorrelation coefficients are calculated on the lines of pixels that are interpolated along that direction (lines 7–9). Finally, the mean of these autocorrelation outputs is calculated (line 10), and the lag of the minimum mean autocorrelation coefficient gives the motion extent along θ (line 11).

In [4], the directional filters, for the calculation of $\Delta_\phi(f)$ and g, derive from the $[-1\ 1]$ kernel rotated and interpolated to fit each direction. The small support of these filters renders them particularly sensitive to additive noise, reducing the robustness of the method. Thus, for SNR < 35 dB, Yitzhaky and Kopeika [4] suggest avoiding the calculation of g and computing the ACF directly on $\Delta_\theta(f)$. In that case, the ACF is significantly affected by the image correlation properties. To resolve the above, Sobel filters have been used in this work for the derivative calculations, replacing the $[-1\ 1]$ kernels of [4]. Moreover, for SNR < 40 dB, the system robustness is further increased by applying on the input frame an adaptive Wiener filter [14], which adapts to the local image variance and thus preserves the image edges.

In [4], the frames that are considered involve global motions, which would derive from camera motion. To deal with object motion as well, a masking stage should be included (line 6). The mask should be applied after the derivative g has been calculated. The mask is generated by considering a pair of frames before and after the current frame, avoiding the blending regions at the borders of the object, to minimise the interference from the static background [1].

The computational load of the blur identification process can be significantly reduced by considering fewer directions for calculating $\Delta_\phi(f)$ and also fewer ACF lags. Moreover, by rotating f by θ and then calculating g on the rotated frame, only two filters should be applied for the generation of g: a vertical and a horizontal filter. Such issues are further discussed in Sections 5 and 6.1, where different design options are evaluated with respect to the system's area and throughput, whereas Section 6.2 discusses their impact on the system's performance.

2.2 Assumption validation

In this section, a scheme that validates the initial linear and uniform motion assumption is proposed. This validation process is equivalent to performing a classification where cases of linear and uniform intra-frame motion are classified as positive, whereas all other cases give rise to negative classification. This classification can be done using two types of data: the normalised total intensities $I(\phi)$ and the coefficients of $\bar{R}$.

To demonstrate their properties for different motion types, the above data are calculated on the motion-blurred frames derived after the application of different motion PSFs on the 240×320 ground-truth image of Fig. 2. It should be noted that the example that follows is an introductory example that employs a very simple image. The validation scheme will be extensively evaluated in Section 6.2, where its performance is investigated by employing a large number of tests with various semi-synthetic video data.

Four motion PSFs are applied on the ground-truth image of Fig. 2. These are each associated with a different motion type: (i) a linear and uniform PSF with an extent of 15 pixels, which is applied in the vertical direction (Fig. 3*a*), (ii) the non-uniform linear PSF of Fig. 3*b*, which is also vertically applied, (iii) the non-linear and uniform PSF of Fig. 3*c*, and (iv) the non-linear and non-uniform PSF of Fig. 3*d*. The experiments demonstrated in Fig. 4 are executed after adding Gaussian noise that results in an SNR of 50 dB, since 50 dB is the typical SNR of digital cameras. The intensities $I(\phi)$ and coefficients $\bar{R}$, which are calculated for each blurred frame, are illustrated in the graphs of Fig. 4.

The mean ACF along the motion direction normally has a particular shape for the case of linear and uniform motion, which is similar to that of Fig. 4*a*. Specifically, it contains three dominant lobes: a positive lobe at lag 0 and two negative lobes that are symmetrical with respect to the y-axis. The lag of the minimum coefficients of the two symmetrical negative lobes indicates the motion extent. In Fig. 4*a*, for example, the minimum ACF coefficients lie at lag ± 15. Therefore the estimated motion extent is equal to 15. The relation between the minimum ACF coefficients and the linear motion extent is explained in detail in [4]. Experiments have shown that the more the given ACF diverges from the form that is described above, the more the corresponding motion diverges from the ideal linear and uniform case. Therefore negative classification occurs in the following cases:

1. The total numbers of positive and negative main lobes is different than those stated above, that is one positive lobe at lag 0 and two symmetrical negative lobes. This is the case for high-frequency temporal vibrations that lead to non-uniform motion, where more lobes are formed. Specifically, in that case, at each side of lag 0, there is a positive lobe surrounded by two negative lobes, as can be observed in Fig. 4*b*. Non-linear motions, both uniform and non-uniform, may as well generate multiple irregular lobes, as illustrated in Figs. 4*c* and *d*.

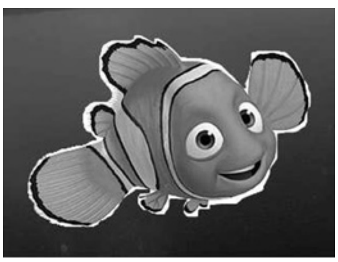

Fig. 2 *Ground-truth image*

2. The lag of the minimum ACF coefficient, that is the position of the minimum ACF coefficient on the x-axis of the ACF graph, is very close to lag 0. This indicates a motion extent that is too small to reflect the actual PSF. This is normally observed in the case of non-linear motions, both uniform and non-uniform, as shown in Figs. 4*c* and *d*.

When a frame is linearly blurred along a particular direction, there is a clear minimum in the TI graph, which indicates the motion direction (Figs. 4*a* and *b*). In the case of non-linear

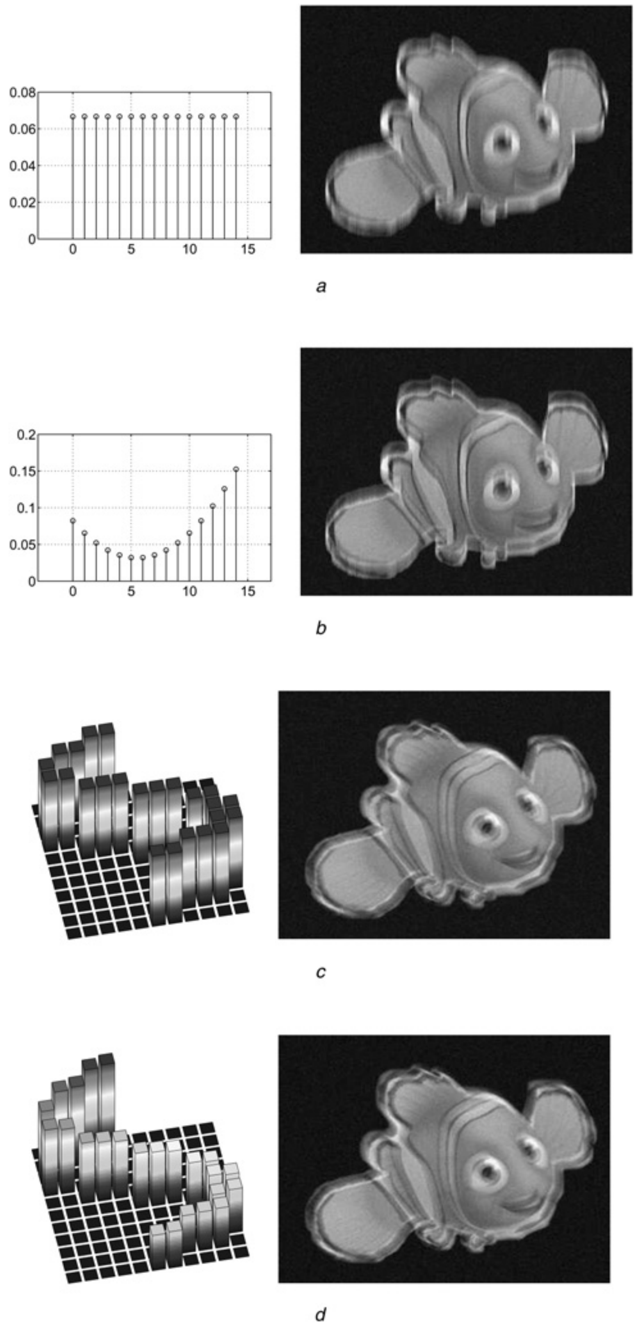

Fig. 3 *Various motion PSFs and the corresponding motion-blurred images*

a, *b* X-axis indicates the pixels, whereas y-axis gives the PSF weights
c, *d* Pixels are represented as squares with their 'facades' corresponding to the PSF weights
a Linear and uniform motion
b Linear and non-uniform motion
c Non-linear and uniform motion
d Non-linear and non-uniform motion

doi: 10.1049/iet-cdt.2009.0053

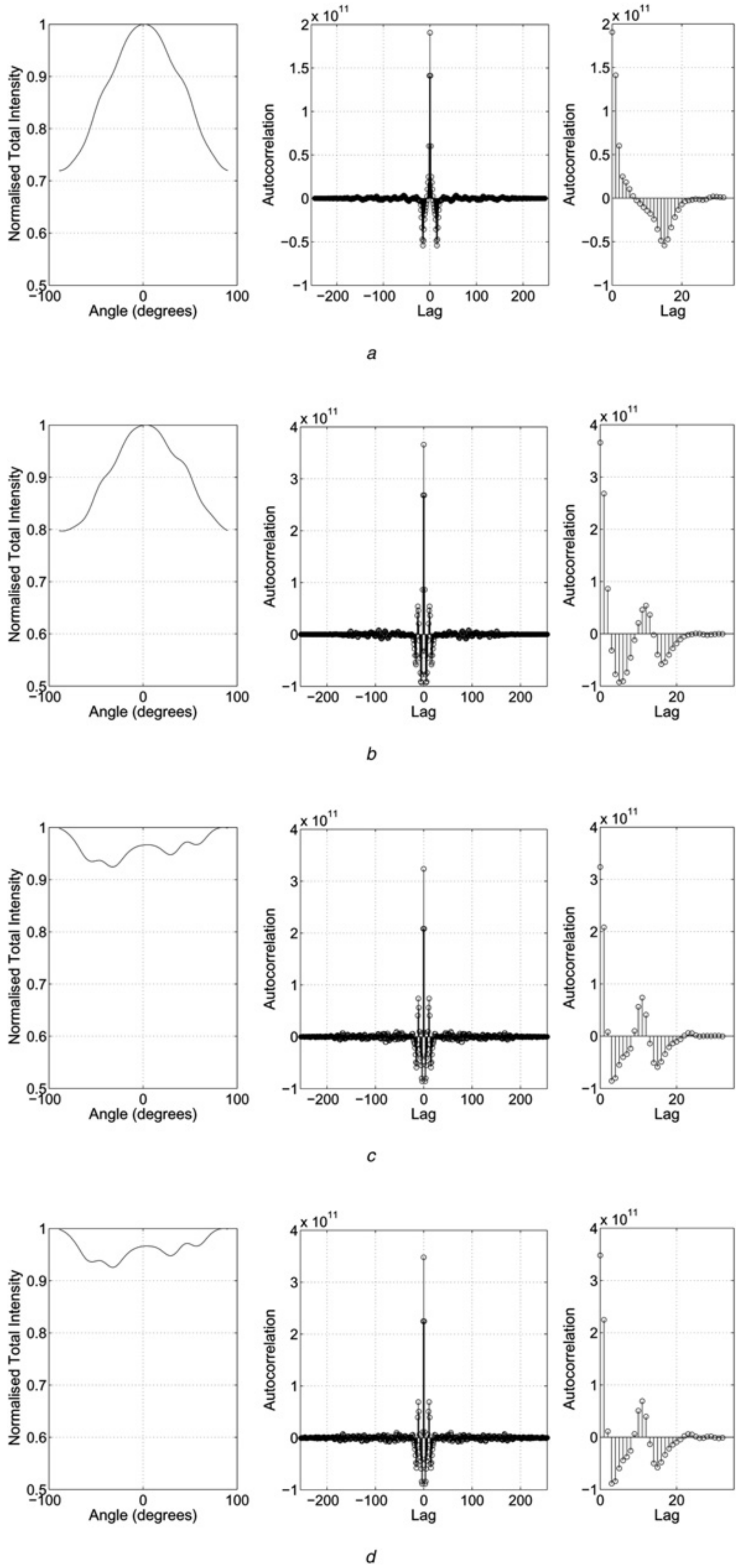

Fig. 4 *Calculated normalised total intensities* $I(\phi)$ *and autocorrelation coefficients* $\bar{R}$ *for the indicated motion types*

a Linear and uniform motion
b Linear and non-uniform motion
c Non-linear and uniform motion
d Non-linear and non-uniform motion

motions, this minimum is not clear (Figs. 4*c* and *d*). Therefore for non-linear motions, the minimum normalised TI is significantly higher (Figs. 4*c* and *d*) than in the case of linear motions (Figs. 4*a* and *b*), and this value is used in the classification.

Linear and uniform motions of very large extents may as well be undesired for the given specifications, owing to the large supports of the associated PSFs, which increase the computational cost and required precision for the subsequent restoration block. By setting a maximum lag in the ACF computations, only linear motions whose extent is smaller than that lag produce negative ACF lobes. Larger motions are automatically classified as negative, which is appropriate for the system that will be described in Section 3.

2.3 Inter-frame and intra-frame motion

Owing to the continuity of motion in subsequent frames, for a linearly moving object, the intra-frame motion PSFs should be consistent, in both extent and direction, with the inter-frame motion vectors. Thus, if the intra-frame and inter-frame motion parameters are not consistent, the linearity assumption is probably invalid. The above can be incorporated in the validation scheme as a further check for motion linearity, with the extra computational cost required for registration. When the subsequent reconstruction block is SR based, registration is executed anyway for the SR method to be applied [12, 13, 15–17], and thus the above check can be implemented with no additional computational cost.

3 Accounting for intra-frame motion in a system with an adaptive image sensor

In [2], we proposed a video enhancement system that is based on an adaptive image sensor [3, 10]. The system of [2] exploits the space–time trade-off, according to which a decrease in the spatial resolution of an image sensor results in an increase in the temporal resolution of the output video sequence [18, 19]. In particular, the motion regions are configured to larger pixel sizes that produce high frame-rate samples with reduced motion blur. The system of [2] compensates for the decrease in the spatial resolution of the motion areas by applying SR on the sequence of high frame-rate samples. Thus, the final output has high resolution in both space and time. However, for very fast motion, the time samples are inevitably blurred themselves, and the fidelity of the final reconstructed output is bounded by the blur effect.

For the rest of the paper, 'LR' ('HR') refers to low (high) spatial and, thus, high (low) temporal resolution.

In this paper, the system of [2] is extended so as to estimate the accurate PSFs of the LR samples that are produced on the motion areas. This is done by utilising the BIV scheme to estimate the intra-frame motion of the LR samples. By accurately estimating the motion PSFs of the LR frames that participate in the reconstruction process, the SR reconstruction quality is increased. The remainder of this section explains how the reconfigurability of an adaptive image sensor is combined with the BIV scheme to provide accurate intra-frame motion identification and thus improve the output of the subsequent SR reconstruction block.

The adaptive image sensor can be locally configured to form LR areas that produce high frame-rate samples, because of the space–time trade-off [12], thus fragmenting the motion trajectory. Fig. 5 shows the raw outputs for two sensor configurations during the HR integration interval, that is the time required by the elementary pixels to achieve a certain SNR [20, 21]. The number of time samples produced during the HR integration interval increases as the pixel size increases, that is as the spatial resolution decreases, in accordance with the space–time trade-off [18, 19]. Thus, a grid of elementary pixels would render a single, motion-blurred frame (Fig. 5*a*), while a 2 × 2 configuration would give four time samples (Figs. 5*b*1–*b*4), each containing a fragment of the trajectory of Fig. 5*a*. To increase the spatial resolution of the output, the LR samples are fused by applying SR techniques [17]. For SR to be effective, a PSF should be estimated within a certain accuracy for each sample.

The proposed system utilises a simple blur detection block, which is based on the comparison of the strongest edges of the object with those of the background, to identify cases where the current configuration produces samples with negligible motion blur, as in Figs. 5*b*1–*b*4. In such cases, an isotropic Gaussian PSF can be employed to associate each LR pixel to the HR pixels of the underlying HR grid [17]. In all other cases, the Gaussian assumption is inadequate and blur identification is required to estimate the motion parameters. Thus the BIV scheme, as described in Section 2, is executed. If BIV finds the linear and uniform motion assumption invalid, the pixel size increases in the next sensor reconfiguration, to produce samples with reduced, more linear motion blur. In this manner, the initial non-linear and/or non-uniform motion trajectory is fragmented into shorter, more linear parts. If BIV finds the initial motion assumption valid, accurate linear PSFs can be estimated, and the pixel size thus remains constant.

The pixel size of the adaptive sensor depends on the outputs of blur detection and *BIV* blocks, as described above. These blocks comprise a classifier whose binary output c determines the pixel size in the next sensor reconfiguration: If $c = 1$, the pixel size remains constant, as it allows an accurate estimation of the PSF, employing either the Gaussian or the linear and uniform motion

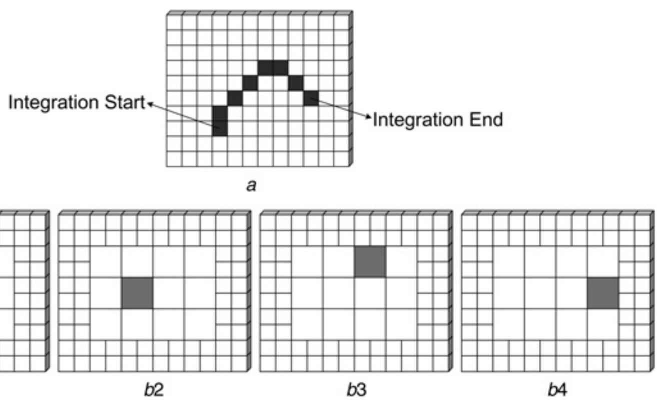

Fig. 5 *Outputs during the HR integration interval for two sensor configurations*

a 1 × 1 configuration
*b*1–*b*4 2 × 2 configuration

doi: 10.1049/iet-cdt.2009.0053

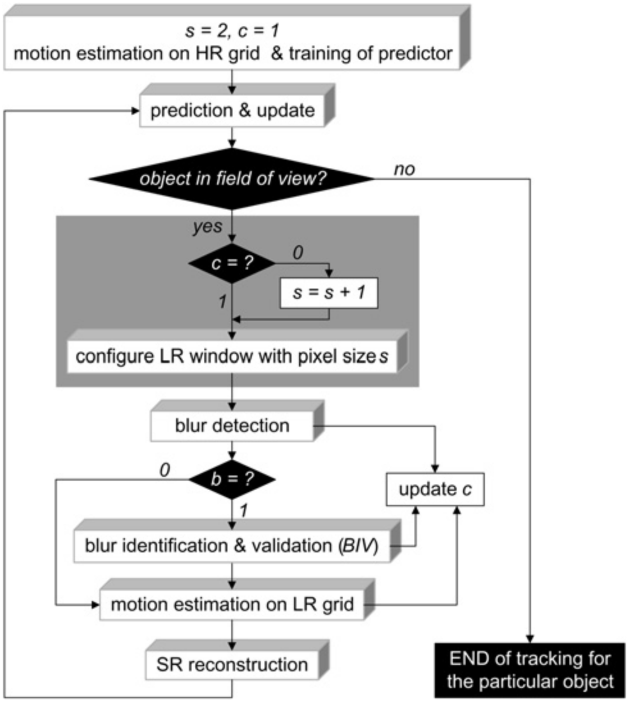

Fig. 6 *Algorithm of the system operation for a dynamic region*

assumption. If $c = 0$, the pixel size increases in the next reconfiguration. A mechanism that reduces the pixel size every N frames can also be accommodated but has not been investigated in this work.

For linear object motion, the non-isotropic PSFs describing the intra-frame motions are expected to be consistent with the inter-frame motion vectors. In case of inconsistency, invalidity of the initial linearity assumption is indicated. The above can be used as an additional validity check and could thus be incorporated into the classifier that is described above, giving $c = 0$ in the case of inconsistency. However, it should be noted that this additional validity check has not been incorporated in the experiments presented in this paper.

Ideally, the adaptive sensor would be reconfigured at every new HR integration. In reality, reconfiguration is sparser, depending on the technology of the given sensor. The proposed video enhancement system operates as follows. Moving objects are detected with a rough motion estimation [22] on the HR grid, as indicated in Fig. 6. For every moving object, the control of Fig. 6 is employed, where s denotes the LR pixel size, with $s = 2$ corresponding to the smallest LR pixel size as demonstrated in Figs. 5b1–b4 and c denotes the binary output of the classifier that determines the change in the pixel size in the next reconfiguration. Moreover, b denotes the binary output of the blur detection block; $b = 0$ indicates negligible motion blur, for which the Gaussian assumption is adequate and BIV is skipped, whereas $b = 1$ indicates that BIV is required. The part of Fig. 6 included in the grey rectangle is executed only in those HR integration intervals when the sensor is reconfigured. A Kalman filter predictor [23, 24] is employed to determine the position of the object in the next HR integration. When sensor reconfiguration occurs, that position determines the location of the LR area, whereas its pixel size depends on the validity of the linearity assumption in the last HR integration, indicated by the value of c. For $c = 0$ the pixel size increases, whereas for $c = 1$ it remains the same. For each LR area, an LR sequence with reduced blur is produced, and the PSFs are estimated, based on the outputs of the blur detection and BIV blocks, as described in the previous paragraphs. The LR samples are registered using a motion estimation block. Motion estimation is implemented using Lucas–Kanade optical flow [22] and Shi–Tomasi good feature extraction [25]. The PSFs corresponding to the LR samples are then used by the *SR* block that executes the reconstruction on the HR grid [17]. This produces for each LR area an output with high resolution both in space and time, and thus motion deblurring is locally executed on the dynamic regions of the scene. At every new HR integration, the control starts at the second block of Fig. 6. The loop ends when the particular object exits the field of view.

4 Effect of BIV on SR reconstruction quality

This section investigates how the quality of the reconstructed output is affected by various sensor configurations and PSF

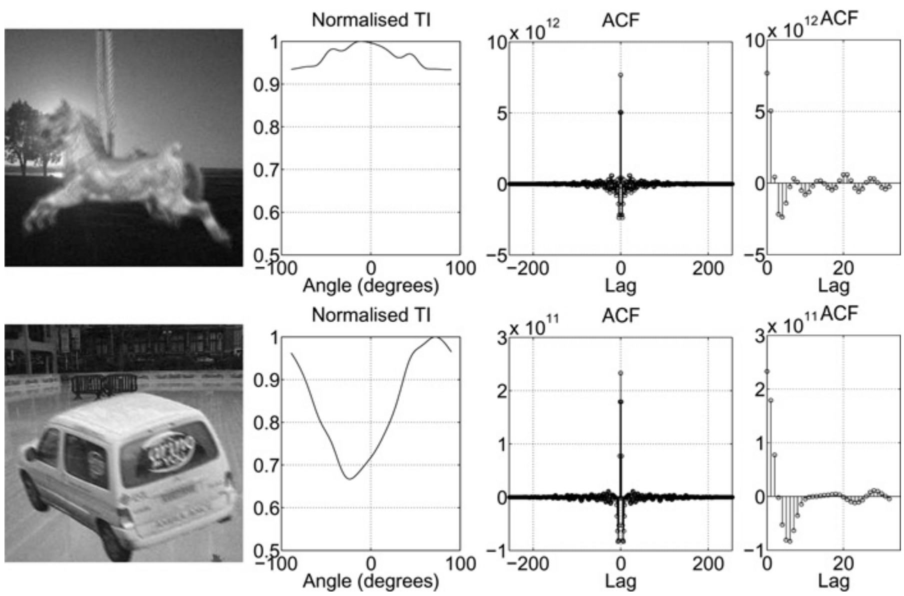

Fig. 7 *TI and ACF for carousel (top) and ambulance (bottom row)*

On the right, a detail of the ACF is presented, for lags 0–33 assumptions, in order to demonstrate the benefits from the utilisation of the proposed joint BIV prior to SR. It should be noted that the current section aims to demonstrate the importance of BIV, by showing how the SR output quality improves when a realistic motion assumption is employed in the reconstruction process. The actual evaluation of the FPGA-implemented BIV method will be presented in Section 6.2, after its hardware architecture has been described in detail.

The evaluation is assessed using semi-synthetic data. That is, a real image, captured with a hand-held digital camera, has been shifted, blurred, downsampled and contaminated with noise, to synthetically produce the LR sequences. Thus, the 'ground-truth' frame is known and is used as a reference to evaluate the reconstruction quality. Various parameters are employed, including the type of PSF assumption, the LR pixel size and the noise level. The input frames have been contaminated with white Gaussian noise resulting to a range of SNRs between 10 and 50 dB. The SNR level is defined as follows: $\mathrm{SNR} = 10\log(\sigma_f^2/\sigma_n^2)$, where σ_f denotes the standard deviation of the noise-free image and σ_n denotes the noise standard deviation. The iterative SR approach of [17] is used, and 30 iterations are executed for each estimation. To exclude any evaluation errors owing to the blending of the object with the background, SR is applied on the isolated foreground objects. The number of frames produced during HR integration for each pixel size is subject to the space–time trade-off [2]. If this number is k for the current configuration, k additional neighbouring frames are used in SR, for increased robustness [17].

Fig. 7 shows the TI and ACF outputs of BIV (Section 2) for two moving objects with non-negligible motion blur ($b = 1$). Both cases employ 2×2 configuration and 256×256 LR resolution. The frame on the left is one of the four samples generated during HR integration at 50 dB SNR. According to the validation criteria described in Section 2, the TI and ACF outputs of Fig. 7 indicate that for the 2×2 configuration the linearity and uniformity assumption is

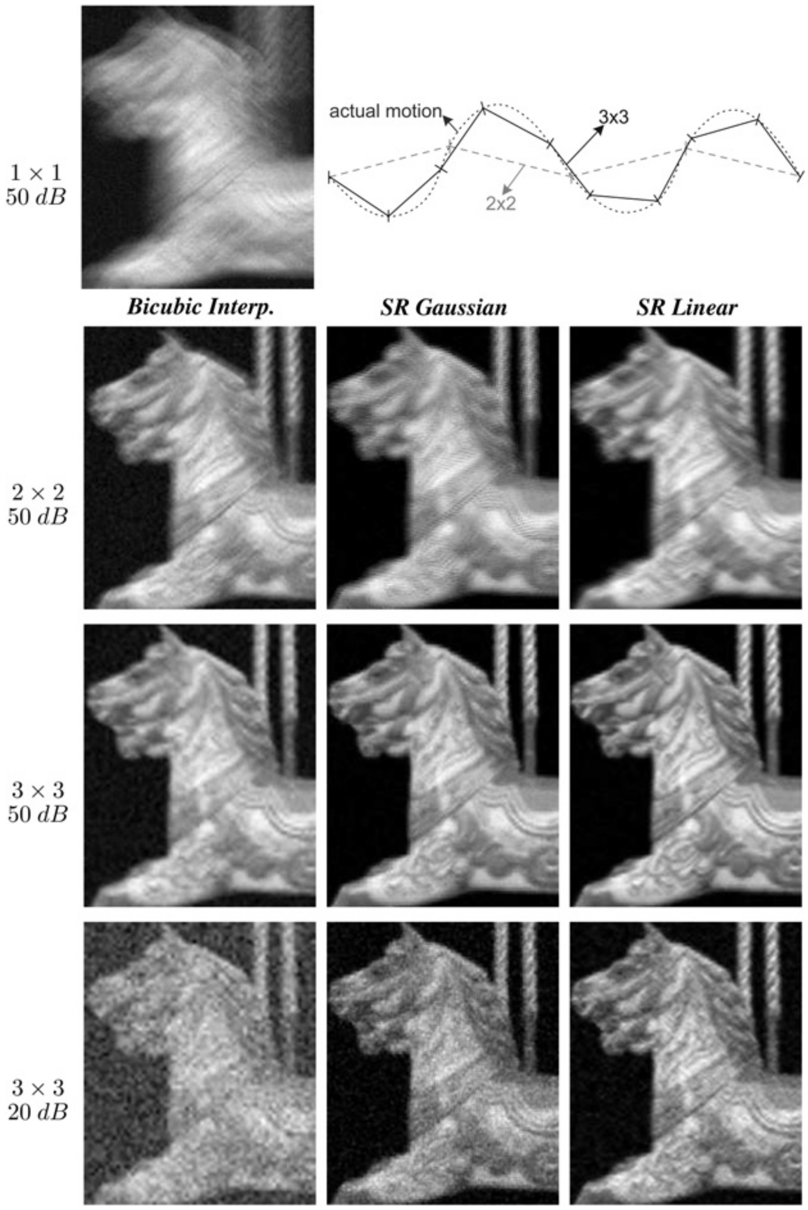

Fig. 8 *Reconstructed output for carousel for the indicated sensor configurations, noise levels, and reconstruction methods*

Row 1: Raw output for configuration 1×1 (elementary pixel grid), and the intra-frame motion of the time samples produced during HR integration for each configuration

Rows 2–4: SR reconstructed outputs when employing the indicated PSF approximations

IET Comput. Digit. Tech., 2011, Vol. 5, Iss. 4, pp. 271–286
doi: 10.1049/iet-cdt.2009.0053

valid for *ambulance* and invalid for *carousel*. Indeed, the actual motion of *carousel* is not sufficiently linear for a 2 × 2 configuration, as the trajectory shows in Fig. 8 (top right). Therefore a 3 × 3 configuration is employed in the next reconfiguration of the adaptive image sensor.

The detailed images of Fig. 8 show the SR-reconstructed output for *carousel*, for 2 × 2 and 3 × 3 pixel configurations. For reference purposes, bicubic interpolation is applied on a single LR frame, with magnification factors 2 and 3, respectively. The SR output is given both for Gaussian PSF approximations, whose support corresponds to the LR pixel size [13, 17], and for linear PSFs estimated by BIV. The validation block of BIV indicates that the linear motion assumption is invalid for the 2 × 2 configuration and valid for the 3 × 3 configuration. The system output is that of the 3 × 3 configuration with SR that uses linear PSFs (Fig. 8).

The last row of Fig. 8 demonstrates the system robustness, presenting the reconstructed outputs for significantly noisy LR samples (SNR = 20 dB). Fig. 9*a* quantifies the evaluation giving the root mean square error (RMSE) values with respect to the ground-truth, for the above scenarios and SNR from 10 to 50 dB.

Contrary to *carousel*, *ambulance* passes the validity check for 2 × 2 configuration (Fig. 7); thus, the pixel size remains at 2 × 2. Fig. 10 presents the indicated outputs for 50 and 20 dB. The system output is that of SR with the linear PSF approximation. The associated errors are given in Fig. 9*b*. It can be concluded that when the linear PSF is estimated, the SR output improves compared to the use of a Gaussian PSF.

The above evaluation demonstrates that when the linear PSFs, which describe the intra-frame motions of the LR frames participating in SR reconstruction, are estimated and taken into account in the reconstruction process, the SR output improves dramatically compared to the case where a naive assumption of Gaussian PSFs is made. In the remainder of the paper, an efficient real-time hardware architecture of the proposed blur identification and classification scheme is proposed, implemented on FPGA and evaluated.

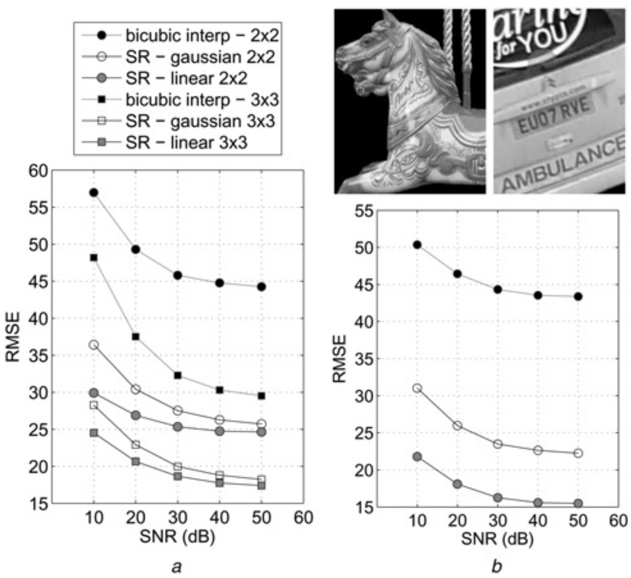

Fig. 9 *Errors for the two sets of experiments, for various SNRs*

a errors for *carousel*
b errors for *ambulance*
Legend applies to both graphs, with *b* containing only the 2 × 2 configuration values. Ground-truth frames are shown at the top

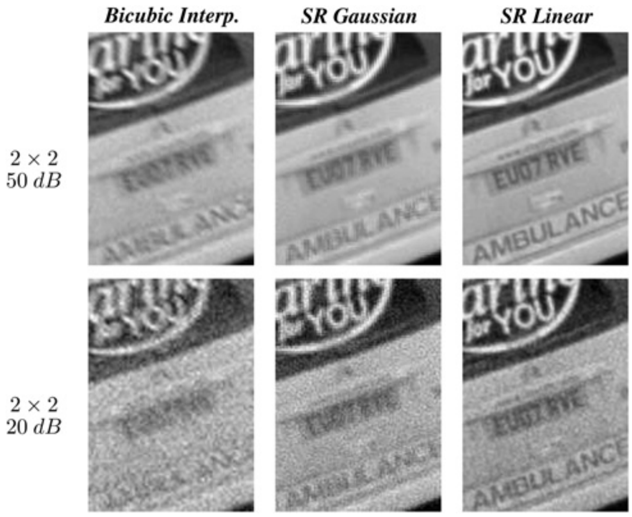

Fig. 10 *Reconstructed outputs for ambulance*

5 Joint identification and validation: hardware architecture

The block diagram of the proposed FPGA-based architecture, which implements the algorithm of Section 2, is presented in Fig. 11. The proposed architecture is fully pipelined and optimised with respect to throughput, area and data reuse. The input frames and the corresponding binary masks are read from a single-port off-chip random-access memory (RAM), the Image Memory. The processing is mainly divided into two stages, each associated with certain system blocks and input stream, as explained in the following subsections. The first processing stage identifies the motion direction θ, and the second stage calculates the motion extent L and implements the validation scheme.

5.1 Directional filters and minimum TI block

During the first processing stage, the input frame is horizontally traversed, and the directional image derivatives $\Delta_{\phi}(f)$ and corresponding total intensities $I(\phi)$ are calculated (Section 2). To maximise data reuse, an extract processing window (EPW) block produces a processing window in every clock cycle. In every cycle, all the elements of the current processing window are fed in parallel to the directional filters, for the computation of the directional image derivatives, thus maximising the parallelism in the derivative calculations.

The Sobel approximation to the derivative is employed, which is in 0° as follows

$$S_{0^{\circ}} = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} \quad (1)$$

In directions other than the horizontal and vertical, the Sobel approximation requires 5 × 5 coefficients, to accommodate the corresponding rotations of the derivative kernel.

As explained in Section 2, the image derivatives should be calculated on the umasked frames. After the derivatives have been calculated, the minimum total intensity (MinTI) block computes for each derivative the sum of absolute intensities only of pixels with mask bit 1. In this manner, the static background is excluded from the intensity calculations. The MinTI block finally returns the direction θ corresponding to

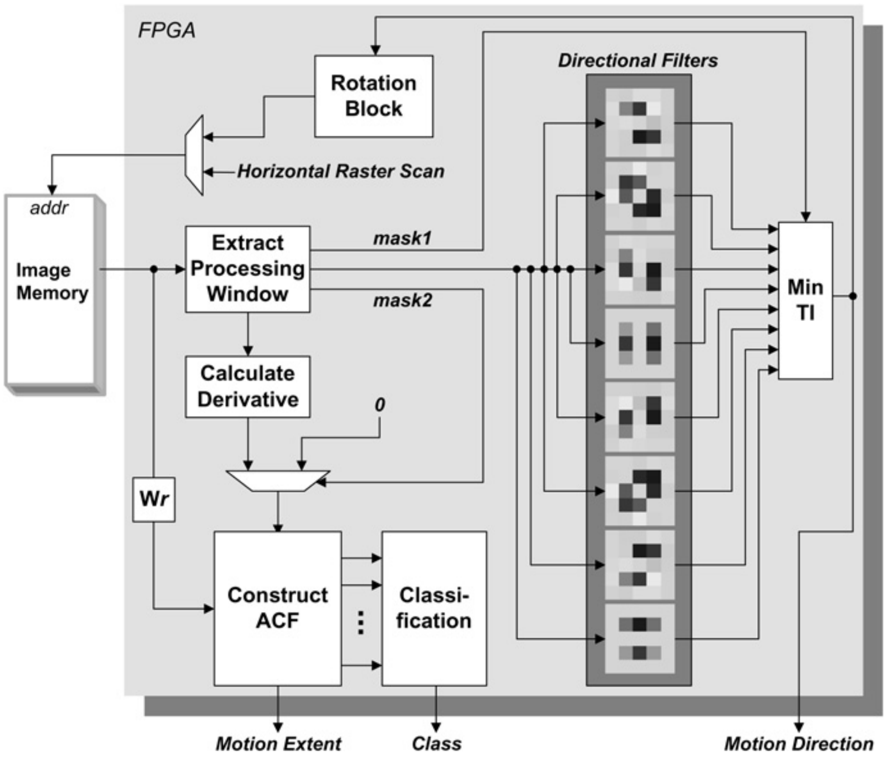

Fig. 11 *Overview of the joint BIV system, as implemented on FPGA*

the minimum TI. This does not necessarily coincide with a filter's direction. That is, if the difference between the minimum TI and the TI corresponding to an adjacent filtering angle are similar, then the block returns the mean direction. The ACF will then be computed by the subsequent blocks along that direction.

Only the motion line matters in the above computations, and not which way the object moves along that line. Therefore the rotation angles included in the filter-bank need to span only half of the Cartesian plane. Fig. 11 demonstrates a filter-bank employing eight different angles; thus, a step of 22.5° is used for the filter formation.

5.2 Rotation block

The mean ACF should be computed on the derivative g in direction θ (Section 2). To do this, instead of combining horizontal scanning with non-horizontal processing, it is computationally more efficient to do the opposite: traverse the frame along θ, thus horizontally scanning the rotated frame and processing the corresponding rows. In addition, by doing the above, only two directional filters need to be available for the generation of g: a vertical and a horizontal filter.

The rotation is implemented using the nearest neighbourhood interpolation scheme, which transforms the rotation problem to the simpler problem of generating the corresponding stream of RAM addresses. Thus, the rotation block takes θ as its input and generates the streaming addresses for the horizontal raster scan of the rotated by θ frame. Lookup tables return the sine and cosine values corresponding to θ, for the formation of the Cartesian rotation equations

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{2}$$

The streaming addresses are generated by the rotation block by employing nearest-neighbour interpolation, that is by simply selecting the nearest pixel after applying (2).

5.3 EPW block

This block increases data reuse by producing in every cycle a convolution window. The block employs first in first out (FIFOs) and registers in the circular buffering scheme demonstrated in Fig. 12. FIFOs are implemented with on-chip dual-port Block RAM (BRAMs). The block operates in two modes, associated with the two main processing stages, thus saving hardware resources.

The first mode, Mode 1, which is denoted in Fig. 12 with black shade, is associated with the generation of the directional image derivatives $\Delta_\phi(f)$, as has been mentioned in Section 5.1. In Mode 1, the input frame f is read into the buffers in horizontal raster scan order. Thus, the active length of the FIFOs equals the length of the rows of f. All four FIFOs feed the same window of registers, whose size is 5×5, that is the maximum required filter size (Section 5.1). As mentioned in Section 5.1, in every cycle, the processing window is read in parallel by the directional filters of Fig. 11.

The second mode, Mode 2, is associated with the calculation of the image derivative g. Therefore, in Mode 2, the buffers of Fig. 12 are fed with f rotated by θ, and an active FIFO width equal to the diagonal of frame f is required to accommodate the longest possible row. Two independent buffering structures, related to different filtering operations, are formed. Each includes a FIFO pair and a separate window, denoted in Fig. 12 with a white shade. The first filtering feeds a vertical Sobel filter that calculates the vertical image derivative $\Delta_{\theta+90^\circ}(f)$, to remove object edges as explained in Section 2. The vertical derivative is directly consumed by the second buffering structure, which feeds a horizontal Sobel filter, and the

IET Comput. Digit. Tech., 2011, Vol. 5, Iss. 4, pp. 271–286
doi: 10.1049/iet-cdt.2009.0053

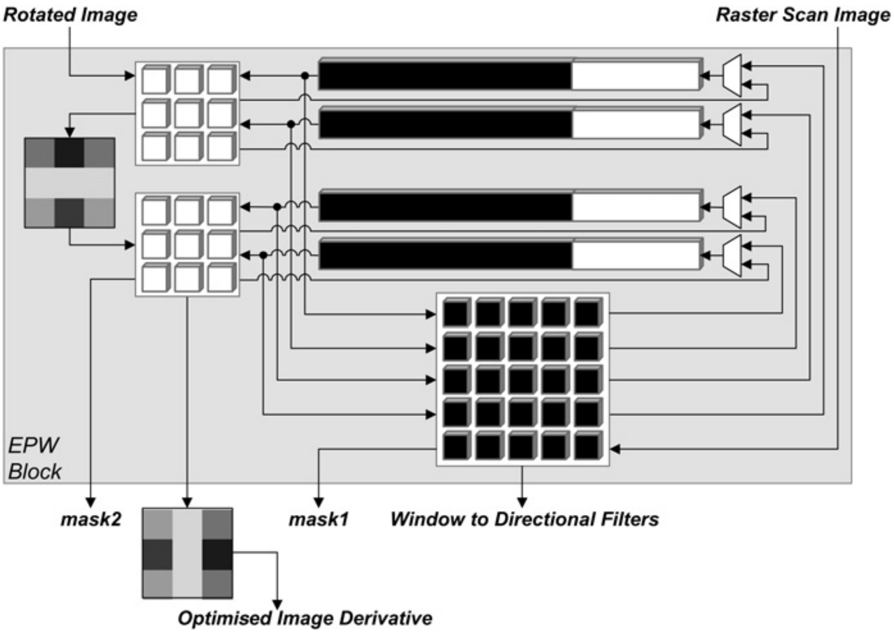

Fig. 12 *EPW block*

The block operates in two different modes, which are distinguished in the figure with the black and white shades. Each mode is associated with a processing stage

image derivative g is produced. Only 0° and 90° derivatives are, therefore, involved in *Mode 2*, and a 3×3 window size suffices for the Sobel approximation.

As stated in Section 5.1, the derivatives are applied on the unmasked frame; thus, the mask bits are propagated to the next processing levels along with the corresponding pixel values.

5.4 Construct ACF and classification blocks

The ACF calculations are executed on the rows of the image derivative g, which are generated in a streaming manner – one pixel per cycle – by the structure of Fig. 12. To minimise the latency and resources, redundancies involved in the ACF computation are removed. Therefore the ACF coefficients are computed only for half of the Cartesian plane, since the ACF is symmetrical with respect to axis y (Fig. 4). In addition, since the expected intra-frame motion extent is smaller than the row length, only ACF coefficients up to a predefined maximum lag are calculated.

The construct ACF block of Fig. 13 consumes one pixel per cycle. To clarify its functionality, Fig. 14 demonstrates the dataflow for a hypothetical six-pixel row, using a maximum lag of three pixels. Each register of the two groups of Fig. 13 corresponds to a particular lag (0–3), as indicated by the indexing of Fig. 14. The upper rectangle displays the contents of registers Ra at every cycle, while the lower rectangle shows the values that are added to each Rb register. To obtain the mean ACF, the block processes in this manner the entire image, accumulating information from all rows in registers Rb. Thus, at the beginning of a new row only the Ra, and not the Rb registers, are reset. Each row contributes in the mean ACF with weight Wr, which is proportional to the number of non-zero mask bits of the particular row. In this manner, rows containing insignificant part of the moving object do not bias the output. The Wr weights should be available at the beginning of each row's processing; thus, for their calculation, part of the pipeline is circumvented to fetch from an earlier level the corresponding mask bits. In the end, registers Rb, each related to a particular lag, contain the coefficients of the mean ACF.

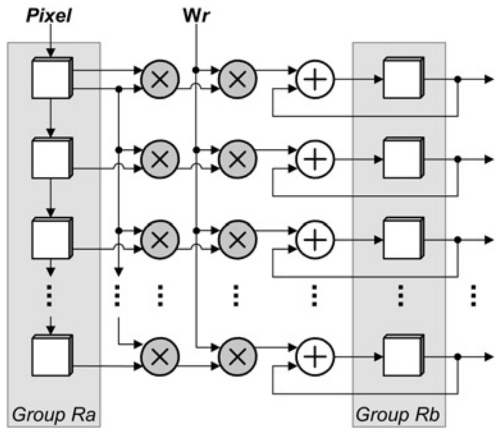

Fig. 13 *Construct ACF block*

The number of registers comprising groups Ra and Rb equals the number of lags under consideration

t	0	1	2	3	4	5
Ra[0]	a	b	c	d	e	f
Ra[1]		a	b	c	d	e
Ra[2]			a	b	c	d
Ra[3]				a	b	c
Rb[0]	a^2	b^2	c^2	d^2	e^2	f^2
Rb[1]		ab	bc	cd	de	ef
Rb[2]			ac	bd	ce	df
Rb[3]				ad	be	cf

Fig. 14 *Time diagram of the ACF computation for a hypothetical six-pixel row 'abcdef'*

t-Axis indicates the clock cycles

The validity of the linear and uniform motion assumption is decided by the classification block. This control block takes the mean ACF coefficients as its input, implements the control logic described in Section 2 and renders a classification bit that indicates if the assumption is valid or not.

6 Hardware evaluation

6.1 Implementation requirements

The design targets a Celoxica ADMXRC4SX board [26], which hosts a Xilinx Virtex-4 FPGA [27] and PL2 zero bus turnaround (ZBT) SSRAM banks. This type of RAM has a pipeline stage at the data and address input lines, which allows it to operate at high clock speeds [26]. The DK5 Handel-C compiler [26] has been used, and the implementation has been placed and routed using Xilinx ISE v.9.2i [27]. The operating frequency of the design on ADMXRC4SX is 120 MHz. The critical path of the circuit lies in the control logic of the rotation block (Section 5).

The total number of clock cycles required to process a single frame is as follows

$$\text{Cycles} = 80 + (V \times U) + (V_\text{r} \times U_\text{r}) + (2 \times G) \qquad (3)$$

where V and U denote the number of rows and columns of the original frame, V_r and U_r denote those of the rotated frame (which depend on θ for given V and U), and G is the number of lags that are considered in the ACF calculation. Also, a constant equal to 80 is because of the reset and latency cycles of the circuit. Thus, the total number of cycles is mainly affected by the frame size and the motion direction (indicated by θ). Specifically, the term $V \times U$ corresponds to the first processing stage when the original frame is horizontally scanned, whereas the term $V_\text{r} \times U_\text{r}$ is related to the second stage, when horizontal scanning is applied on the rotated frame. The number of lags also contributes in a small extent in (3), determining the number of cycles for the construct ACF and classification blocks. On the contrary, the number of directional filters does not affect the number of cycles, as these filters operate in parallel.

The system's throughput is plotted in Fig. 15, with respect to the frame size and the motion direction. The vertical axis of Fig. 15 is in logarithmic scale. To meet real-time requirements, the system should achieve at least 25 fps. All throughput values in Fig. 15 keep a large 'safety margin' above that minimum, the smallest being 38 fps for frame size 1024 × 1024 and ±45°. Owing to its high throughput, the proposed architecture is extremely appropriate for low-power applications, as the operating frequency can be reduced without affecting the real-time performance. The number of external memory accesses is $2 \times V \times U$ per frame, since each frame is read twice from the off-chip RAM, once for every processing stage.

The number of FPGA slices mainly depends on the number of directional filters and is slightly affected by the frame size, as Fig. 16 demonstrates for $G = 30$. The effect of variations of G on the number of slices is relatively insignificant. The number of BRAMs depends on the size of the derivative kernel that is used. For a 3 × 3 derivative filter, whose rotated version requires up to 5 × 5 coefficients, the required number of BRAMs equals 4, for the frame sizes reported in Fig. 16, as for these sizes, the maximum required length of line buffers does not surpass the BRAM length. In addition, six DSP slices are occupied for all image sizes.

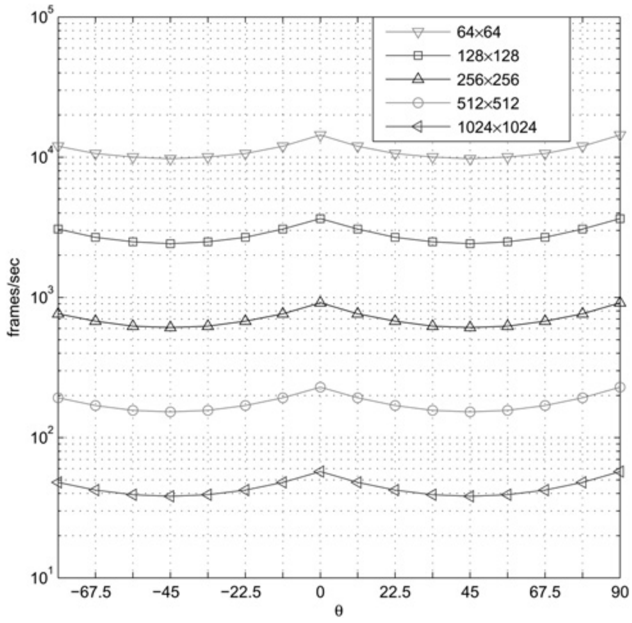

Fig. 15 *System throughput for different frame sizes and motion directions*

Vertical axis is in logarithmic scale

It should be noted that the multiplications associated with the directional filters do not require full multipliers, since they multiply the value of a register to a constant. This is implemented by the Handel-C compiler only with wires that are connected to VCC and ground and (if needed) adders [26]. For example, the multiplication $b = a * 7$ is implemented as $b = (a \ll 3) - a$. Therefore DSP slices are not employed and are thus saved for possible future expansion of the core. As for the adder trees that have been used to compute the final outputs of the directional filters, these are fully pipelined and thus do not affect the operating frequency.

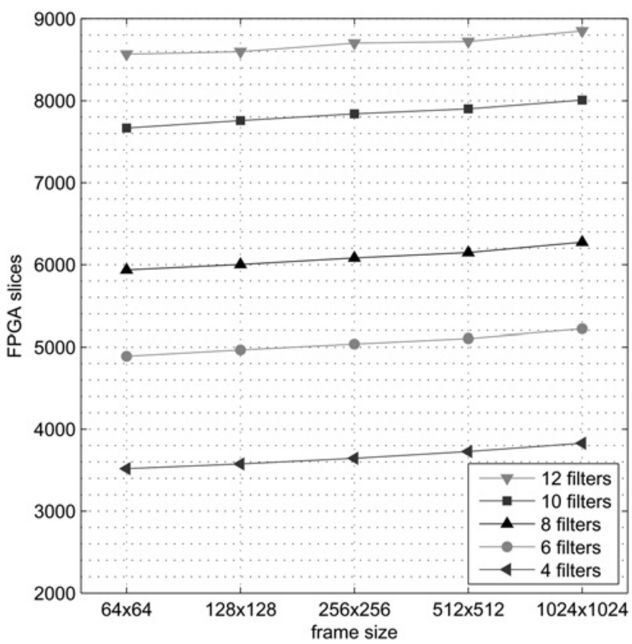

Fig. 16 *Number of FPGA slices for different number of directional filters and frame sizes*

The number of considered ACF coefficients equals 30

doi: 10.1049/iet-cdt.2009.0053

6.2 Performance evaluation

The performance evaluation of the FPGA-based architecture is assessed using semi-synthetic data. That is, real images, which are captured with a simple hand-held digital camera, have been convolved with realistic motion PSFs and contaminated with noise, to synthetically produce the blurred frames. In this way, the system performance is accurately evaluated, since the actual validation class and motion parameters are known, and are compared with the validation output and the estimated parameters of the proposed system. The input frames have been contaminated with white Gaussian noise resulting in a range of SNRs between 10 and 70 dB.

In Fig. 17*a*, a receiver operating characteristic (ROC) curve [28] is used to evaluate the performance of the classifier under different noise levels and employing different numbers of directional filters for the image derivative computations. The input frames used for the generation of the ROC were produced after applying various linear, non-linear, uniform and non-uniform motion PSFs on the ground-truth frames of Fig. 17*b*. Specifically, 50 different motion PSFs of varying spatio-temporal shapes and magnitudes were applied on each ground-truth, locally on the foreground objects of the four leftmost images and globally on the entire rightmost image of Fig. 17*b*. Therefore in the last case, the associated binary mask has ones on the entire area of the unrotated frame *f*. The two first images of Fig. 17*b* are at resolution 256 × 256, whereas the last three are 512 × 512 frames. The entire experiment has been repeated considering four, six and eight directional filters and with SNR 50 and 20 dB. In all experiments, 33 lags have been considered for the autocorrelation calculations, fully accommodating the maximum extent of linear motion that has been used, which is 27 pixels. The ROC curve presented in Fig. 17*a* corresponds to the parameter values $p_1 = 4$ and $p_2 = 8.4$ of the algorithm presented in Section 2, which gave the highest performance in our experiments.

The dotted curve in Fig. 17*a* presents the output of floating point classification, employing full search for the minimum TI, thus a step of 1° and 180 directions. The floating point scenario is implemented in Matlab. The rest of the curves present the outputs of the hardware implementation for numbers of directional filters and noise levels that are indicated in the legend. As observed in Fig. 17*a*, the classifier's performance significantly improves when the number of filters increases from four to six and from six to eight. For the given system that employs Sobel filters as directional filters, for eight directions or more the achieved performance is similar to the best possible floating point full search scenario. Therefore, eight directional Sobel filters are enough to obtain adequate validation performance with respect to motion linearity and uniformity, as the ROC

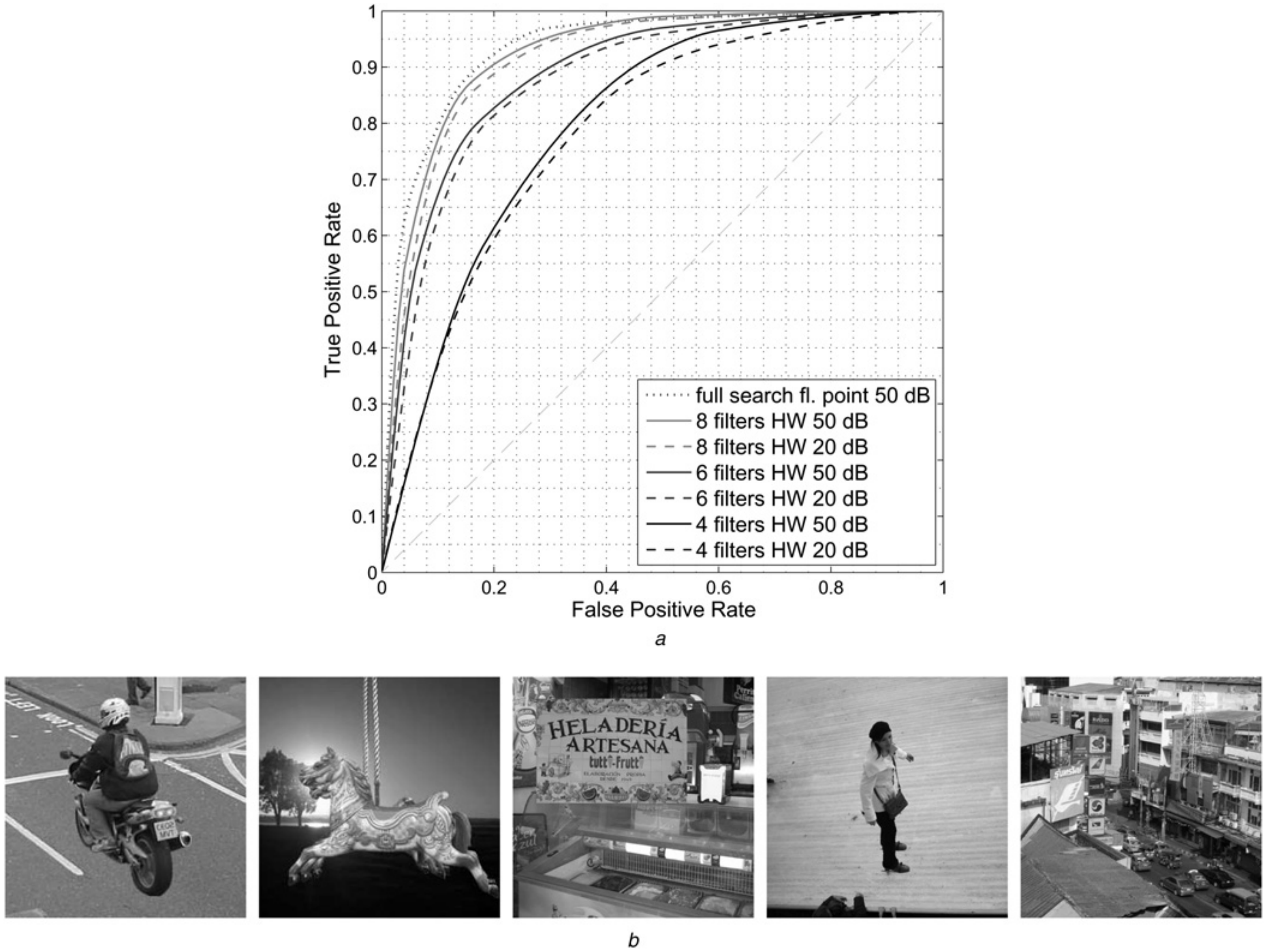

Fig. 17 *Evaluating the performance of the classifier*

a ROC curves for the indicated number of filters and noise level
b Ground-truth frames used for the generation of the test sequences

curve is close to the ROC of the floating point system that uses 180 filters. Fig. 17*a* demonstrates that the variation of the performance for a given number of directional filters under different SNRs is very small, which proves the increased robustness of the classifier with respect to noise.

Six representative examples, taken from the set of experiments used in the above evaluation, are demonstrated in Fig. 18. The scenarios of Fig. 18 involve 50 dB SNR and eight directional filters, while six different motion types are employed: three that are linear and uniform, in Figs. 18*a*, *c* and *e* and three that are not, in Figs. 18*b*, *d* and *f*. In particular, Figs. 18*b* and *d* contain non-linear sinusoidal motion, whereas Fig. 18*f* contains non-uniform motion resulting from high-frequency temporal vibrations (Section 2). The blurred frames are correctly classified; thus, for Figs. 18*a*, *c* and *e*, the validation output is positive, whereas for Figs. 18*b*, *d* and *f* it is negative. The actual parameters of the linear and uniform motions of Figs. 18*a*, *c* and *e* are: extents of 11, 8 and 20 pixels, and directions of -25°, 0° and 45°, respectively. As observed in Fig. 18, these parameters are accurately identified, based on the lag of the minimum ACF coefficient and the minimum TI, respectively.

Fig. 19 presents the outputs of the system for the same scenarios as Fig. 18, but under a high level of noise. In particular, an SNR of 20 dB is employed, and the corresponding noisy frames are shown in Fig. 19. For the examples of Fig. 19, both the classification outputs and the estimated motion direction are unaffected by the heavy noise. A few very small divergences are observed for the estimated motion extents in the case of positive classification. Specifically, for Fig. 19*a*, the estimated motion extent is now 10 pixels instead of 11, which is the actual value and which was accurately identified at 50 dB. In Fig. 19*c*, the extent of eight pixels is identified. However, the high level of noise makes the negative lobe of the ACF less steep and thus the minimum less clear. The identification accuracy of the method is systematically evaluated in the next paragraph.

In Fig. 20, the accuracy of the identified linear parameters is evaluated with respect to the number of directional Sobel filters and the level of noise, for a set of frames that are

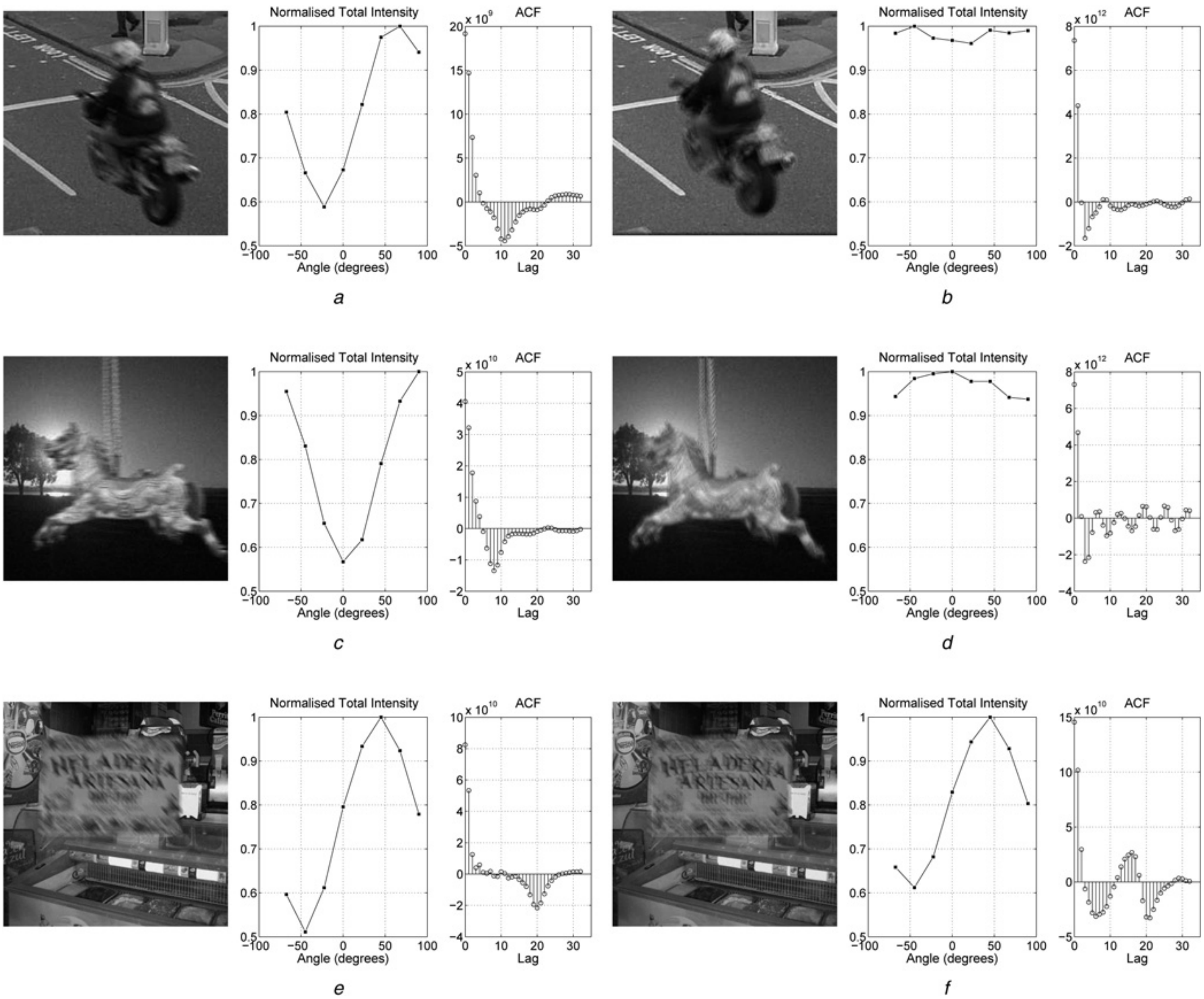

Fig. 18 *Testing examples*

a, *c*, *e* Experiments with valid linear and uniform motion assumption
b, *d*, *f* Experiments with invalid linear and uniform motion assumption
In particular, the motion is non-linear for Figs. 18*b* and *d* and linear but non-uniform in Fig. 18*f*. For each experiment, the calculated TI graph and the mean autocorrelation coefficients are demonstrated in the figure

doi: 10.1049/iet-cdt.2009.0053

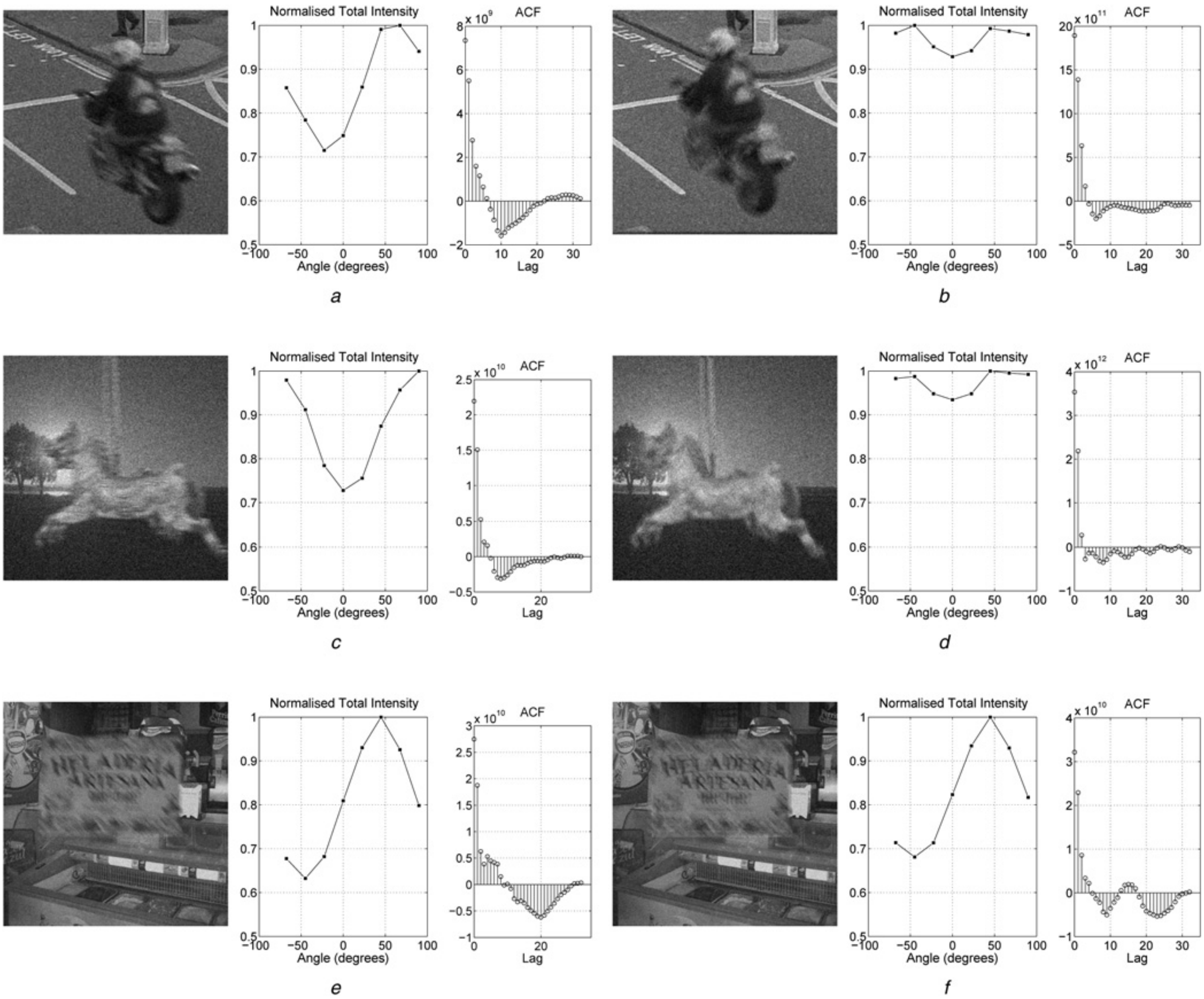

Fig. 19 *Same scenarios as Fig. 18, but under significant noise: SNR = 20 dB*

a, *c*, *e* Experiments with valid linear and uniform motion assumption
b, *d*, *f* Experiments with invalid linear and uniform motion assumption

blurred with linear and uniform motions. To produce this set, 50 different linear motions, of random angles and extents between 7 and 27 pixels, were applied on the ground-truth frames of Fig. 17*b*. The graphs of Fig. 20 present the average values of the errors between the actual and estimated motion parameters (θ, L). The value 180 on the axis with the numbers of filters indicates the output for the full-search, floating point scenario, where a step of 1° covers all integer directions, calculating 180 derivatives. For SNR > 20 dB, the performance of the method is stable, indicating its robustness to noise. Contrary to the validation block, which achieves maximal performance with eight directional filters (Fig. 17*a*), for the identification block to obtain accuracy similar to the full-search floating point scenario, more than eight filters need to be used. This is expected, since validation is a binary problem, but blur identification is not. Thus, the choice of the number of filters for joint BIV depends on the required accuracy in the estimation of motion direction and extent, which is imposed by the given application and mainly the subsequent reconstruction block that uses these parameters.

Utilising the output of the assumption validation process can significantly improve the quality of subsequent restoration, as has been demonstrated in Section 4.

7 Conclusions and future work

This paper proposes a joint BIV scheme, which not only estimates the linear blur parameters, as achieved by traditional blur identification methods, but also validates the initial linear and uniform motion assumption. This keeps the computational cost of blur identification low, owing to the use of the linearity and uniformity assumption for the simplification of the ill-posed identification problem, while identifying cases where the assumption is invalid and thus the PSF estimation is inaccurate. In addition, this work proposes a methodology for combining the above scheme with the reconfiguration property of an adaptive image sensor, in order to tackle cases where the initial motion assumption is found invalid. The sensor grid is adapted to the local motions depending on the validity of the linear and uniform motion assumption. In this manner, the appropriate configuration is applied, and thus the optimal PSF approximation is employed

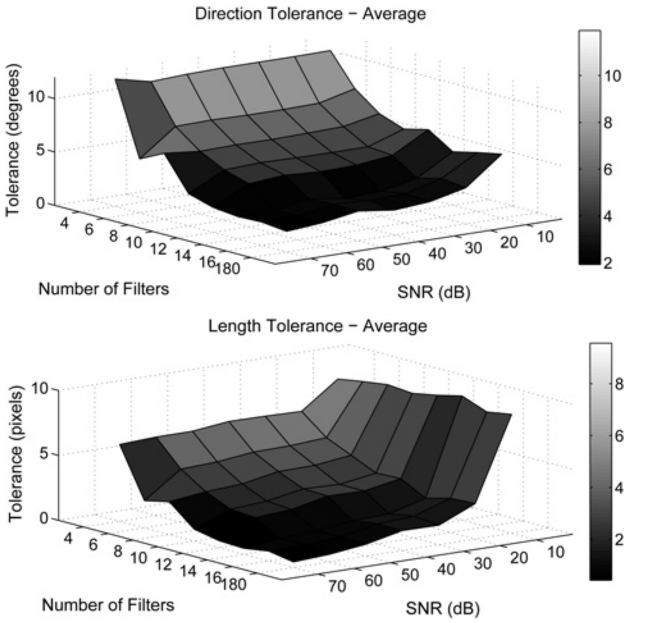

Fig. 20 *Average values of the errors between the actual and estimated motion parameters, for linear and uniform motion*

in SR, improving the reconstruction of the output. Results demonstrate that the proposed methodology surpasses limits of traditional SR, proving the significance of adapting the sensor grid to the given motion.

To target real-time restoration, the proposed joint BIV system is implemented on an FPGA and is evaluated for various sets of parameters. The system's throughput, which mainly depends on the frame size, is significantly higher than the 25 fps real-time requirement, for frame sizes up to 1024 × 1024. The area of the circuit mainly depends on the number of directional filters used for the derivative calculations, while the number of on-chip memories depends on the size of the derivative kernel. The proposed method is robust to noise for SNR as low as 20 dB and correctly validates the initial assumption using eight directional Sobel filters.

Future work includes extending the reconstruction to non-rigid motions and investigating the alternative option of employing second-order blur approximations when the linearity assumption is found invalid. The latter will be evaluated with respect to the related computational costs and hardware resources, and compared to the option of utilising the reconfiguration property of an adaptive image sensor, which has been examined in this paper.

8 References

1 Ben-Ezra, M., Nayar, S.K.: 'Motion-based motion deblurring', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2004, **26**, (6), pp. 689–698

2 Angelopoulou, M.E., Bouganis, C.-S., Cheung, P.Y.K.: 'Video enhancement on an adaptive image sensor'. Proc. IEEE Int. Conf. on Image Processing (ICIP), San Diego, California, USA, October 2008, pp. 681–684

3 Foveon, Inc.: 'Foveon X3 image sensor', www.foveon.com, June 2009

4 Yitzhaky, Y., Kopeika, N.S.: 'Identification of blur parameters from motion blurred images', *Graph. Models Image Process.*, 1997, **59**, (5), pp. 310–320

5 Tekalp, A., Kaufman, H., Woods, J.: 'Identification of image and blur parameters for the restoration of noncausal blurs', *IEEE Trans. Acoust. Speech Signal Process.*, 1986, **34**, (4), pp. 963–972

6 Chen, L., Yap, K.-H.: 'Efficient discrete spatial techniques for blur support identification in blind image deconvolution', *IEEE Trans. Signal Process.*, 2006, **54**, (4), pp. 1557–1562

7 Moghaddam, M.E., Jamzad, M.: 'Motion blur identification in noisy images using mathematical models and statistical measures', *Pattern Recogn.*, 2007, **40**, (7), pp. 1946–1957

8 Chang, M.M., Tekalp, A.M., Erdem, A.T.: 'Blur identification using the bispectrum', *IEEE Trans. Signal Process.*, 1991, **39**, (10), pp. 2323–2325

9 Cannon, M.: 'Blind deconvolution of spatially invariant image blurs with phase', *IEEE Trans. Acoust. Speech Signal Process.*, 1976, **24**, (1), pp. 58–63

10 Constandinou, T.G., Degenaar, P., Toumazou, C.: 'An adaptable foveating vision chip'. Proc. IEEE Int. Symp. on Circuits and Systems (ISCAS), Island of Kos, Greece, May 2006, pp. 3566–3569

11 Kundur, D., Hatzinakos, D.: 'Blind image deconvolution', *IEEE Signal Process. Mag.*, 1996, **13**, (3), pp. 43–64

12 Stern, A., Porat, Y., Ben-Dor, A., Kopeika, N.S.: 'Enhanced-resolution image restoration from a sequence of low-frequency vibrated images by use of convex projections', *Appl. Opt.*, 2001, **40**, (26), pp. 4706–4715

13 Irani, M., Peleg, S.: 'Improving resolution by image registration', *CVGIP Graph Models Image Process.*, 1991, **53**, (3), pp. 231–239

14 Lim, J.S.: 'Two-dimensional signal and image processing' (Prentice-Hall, 1990, 1st edn.)

15 Farsiu, S., Robinson, D., Elad, M., Milanfar, P.: 'Advances and challenges in super-resolution', *Int. J. Imag. Syst. Technol.*, 2004, **14**, (2), pp. 47–57

16 Stark, H., Oskoui, P.: 'High-resolution image recovery from image-plane arrays, using convex projections', *J. Opt. Soc. Am. A*, 1989, **6**, (11), pp. 1715–1726

17 Angelopoulou, M.E., Bouganis, C.-S., Cheung, P.Y.K., Constantinides, G.A.: 'Robust real-time super-resolution on FPGA and an application to video enhancement', *ACM Trans. Reconfig. Technol. Syst.*, 2009, **2**, (4), pp. 22:1–22:29

18 Farrell, J., Xiao, F., Kavusi, S.: 'Resolution and light sensitivity tradeoff with pixel size'. Proc. SPIE Electronic Imaging, San Jose, CA, USA, February 2006, vol. 6069, pp. 211–218

19 Chen, T., Catrysse, P., Gamal, A.E., Wandell, B.: 'How small should pixel size be?'. Proc. SPIE Sensors and Camera Systems for Scientific, Industrial and Digital Photography Applications, January 2000, Vol. 3965, pp. 451–459

20 Gamal, A.E., Eltoukhy, H.: 'CMOS image sensors', *IEEE Circuits Devices Mag.*, 2005, **21**, (3), pp. 6–20

21 Tian, H., Fowler, B., Gamal, A.E.: 'Analysis of temporal noise in CMOS photodiode active pixel sensor', *IEEE J. Solid-State Circuits*, 2001, **36**, (1), pp. 92–101

22 Bouguet, J.-Y.: 'Pyramidal implementation of the Lucas Kanade feature tracker: description of the algorithm' (Microprocessor Research Labs, Intel Corporation, 2002)

23 WELCH G., BISHOP G.: 'An introduction to the Kalman filter', http://www.cs.unc.edu/welch/media/pdf/kalman_intro.pdf, 2006

24 Liu, Y., Bouganis, C.-S., Cheung, P.Y.K.: 'Efficient mapping of a Kalman filter into an FPGA using Taylor expansion'. Proc. Int. Conf. on Field Programmable Logic and Applications (FPL), Amsterdam, Netherlands, August 2007, pp. 345–350

25 Shi, J., Tomasi, C.: 'Good features to track'. Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Washington, DC, USA, June 1994, pp. 593–600

26 Celoxica: http://www.celoxica.com, June 2009

27 Xilinx, Inc., San Jose: http://www.xilinx.com, June 2009

28 Fawcett, T.: 'An introduction to ROC analysis', *Pattern Recogn. Lett.*, 2006, **27**, (8), pp. 861–874

doi: 10.1049/iet-cdt.2009.0053