Robust Motion Filtering as an Enabler to Video Stabilization for a Tele-operated Mobile Robot

Romain Chereau and Toby P. Breckon
School of Engineering, Cranfield University, UK

ABSTRACT

An increasing number of inspection and hazardous environment tasks use mobile robotic vehicles manually tele-operated via a live video feed from an on-board camera. The resulting video imagery frequently suffers from vibration artefacts compromising the accuracy and security of operation in addition to the viable duration for human tele-operation. Here we aim to automatically remove these unwanted visual effects using a novel real-time video stabilization approach. Prior work for hand-held and vehicle mounted cameras is ill-suited to the high-frequency, large magnitude (10-15% of image size) vibration encountered on the short wheelbase, non-suspended robotic platforms typically deployed for such tasks.

Without prior knowledge of the robot ego-motion (or vibration characteristics) we develop a novel four stage filtering approach to identify robust Local Motion Vectors (LMV) for Global Motion Vector (GMV) estimation in successive video frames whilst preserving the required real-time responsiveness for tele-operation. Experimental results over a range of tele-operation scenarios show that the method provides both significant qualitative visual improvement and a quantitative reduction in measurable video image displacement (caused by vibration).

Keywords: stabilization, motion estimation, tele-operation, robotics, vibration

1. INTRODUCTION

The use of tele-operated robotic ground vehicles is now commonplace in many hazardous environments across defence, security, rescue and industrial domains [1, 2]. The capability to navigate remotely via wireless communication over a large variety of terrain, under varying environmental conditions, makes them second to none in a wide range of taskings. Common to all such tele-operated vehicles are one or more video drive cameras, via which a human operator carries out such tasks from the relative safety of a stand-off operating distance (typically 400m-1km using standard wireless video transmission). The successful tele-operation of the vehicle relies primarily upon the clarity of this “operator view” of the remote environment the robot is

traversing and the spatial awareness of the operator within this environment. Due to the physical nature of the robotic platforms typically deployed - no mechanical suspension, short wheelbase, low centre of gravity and lightweight (man-portable) construction - and the nature of the typical deployment terrain the resulting video imagery commonly suffers from significant vibration artefacts. These can compromise the accuracy and the security of tele-operated tasks and impact the viable duration for human tele-operation without causing significant visual fatigue [1]. Even small irregularities in a road or concrete plateau within an urban deployment scenario can cause significant vibration in the video imagery due to the small and lightweight nature of the robotic platforms in use. This can impact operator perception of platform motion, relative distances/clearances and overall remote spatial awareness of the remote environment. Here we outline a methodology for the real-time video stabilization from such tele-operated vehicles with these somewhat unique vehicle vibration characteristics.

A novel image driven approach is proposed based on the use of a robust four-stage filtering methodology for motion vector rejection that enables the use of both an efficient block matching approach [3] and a simplistic, thus efficient, global motion estimation model. We extend the block matching methodology of [3] over a multi-scale pyramid [4] to present an efficient, capable approach with minimal visualization delay under general usage conditions (i.e. 2-frame, 80ms at 25fps). This has a wide range of applications in autonomous robotics [5–7], automotive visual sensing [8–14] and general autonomous sensing applications [2, 15–17].

2. PRIOR WORK

In recent years, several video stabilization approaches have been proposed for hand-held cameras [18–20] and more specifically for intelligent transport and autonomous systems [1, 21–24]. In general two different methodologies have been explored: (mechanical) hardware and software based approaches. Originally hardware approaches dominated due to computational requirements of software approaches on contemporary processing devices. More recently, a lack of flexibility and aspects of size/weight have inverted this trend. In

general, electro-mechanical technologies can have significant size, weight and power requirements for use on a small-scale (man-portable) mobile platform whilst purely optical solutions are restricted to low velocity and magnitude motions within a fixed filtering envelope [25, 26]. By contrast, purely software approaches, or Digital Image Stabilization (DIS) systems, can be dynamically adjusted to detected vibration characteristics within minimal size/weight constraints [21, 27]. For example in [28] the authors combine both technologies obtaining an effective solution whilst in recent mobile robot work specific [1, 22, 24] an external motion sensor provides an input to the DIS.

Overall DIS algorithms for vehicle mounted cameras generally follow a distinct three stage pipeline [21, 27]. Firstly, a *motion estimation* stage estimates the spatial transformation between successive video frames. For the case of a moving platform, this spatial transformation contains two different components: the intentional motions (i.e. vehicle ego-motion) and the undesirable vibration (jitter) motions. Secondly, a *motion filtering* stage extracts the vibration component from this combined transformation. Finally, the inverse of identified undesirable component is used for the *motion compensation* stage to produce a stabilized position estimation of each frame and hence by temporal extension the transmitted video feed.

2.1 Motion Estimation & Filtering

The motion estimation step represents the most critical part involving the accuracy and the time complexity of the system. As a consequence, significant prior DIS work concentrates on this [21]. A stereo image system is used in [29] to firstly recover the ground position and then determinate the vehicle pitch. By contrast, [30] developed a signature-based stabilization able to correct vertical shifts corresponding to 10% of the images height. They characterized each image by a signature based on the horizontal edge histogram. A tracking of the variance in these signatures estimates the inter-frame pitch variations. However, most of the solutions developed considers the motion estimation stage as a motion model fitting process. A motion model describing the inter-frame transformations is firstly determined. A set of local inter-frame motions is estimated next. Finally, the models parameters are determined by fitting the local motions estimation to the motion model. Different models have been used depending on the nature of the vibrations: 2D [31], [32, 33], 2.5D [34] or full 3D [35]. In general it can be observed that motion estimation based on high degree models gives better accuracy (as is to be expected) but at significant

computation cost and potential instability [31–35]. By contrast we propose a somewhat simpler model that, whilst risking being naively considered as ill-suited to the task, can be further enabled by a novel approach of robust motion filtering. Here we put the available limited computational resource, defined by the real-time requirement, into motion filtering instead of (as in prior work) a complex motion model arrangement.

To estimate a set of localized inter-frame motion vectors (Local Motion Vectors - LMV), two generalized strategies are used: a differential approach and a matching approach. The differential approach calculates the optical flow to estimate inter-frame motions of each pixel relying on an image brightness constancy assumption [24, 36, 37]. This approach is time consuming and not robust to the aperture problem (see Section 4.2). On the other hand, the matching approach is based on correlated scene content. This latter approach is divided into two groups: block matching [18, 33] and features matching [31, 35]. Block matching may be time consuming and is limited to translation motions while feature matching can cope with translation, rotation and scale motion variations. However, the robustness relies heavily on the feature selection and [38] exploits scale-invariant feature points providing reliable local features to track. By priming the motion model with a set of robustly filtered LMV we present an over-determined algebraic system, largely clear of significant motion vector outliers, from which a least squares variant can recover our motion estimation [19].

In most stabilization applications, the undesirable (vibration) and desirable (ego) motion components can be distinguished by their relative temporal frequency [21, 27]. As the vibration motion has a higher frequency than the ego-motion of the camera/vehicle, low-pass filters are used to eliminate the unwanted motions (e.g. [39]). Many recent studies [20, 24, 35, 40–43] obtain good results by utilizing a Kalman filter to separate the two components. However, this approach is limited to small oscillations and can introduce a delay into the processing pipeline - a major issue for tele-operation.

3. CHARACTERIZATION OF VIDEO IMAGE VIBRATION

The mobile robot platform considered is specifically designed for Explosive Ordnance Disposal (EOD) operations. This non-suspension tracked-drive vehicle is characterized by a short wheel-base ($\sim 50\text{cm}$), low centre of gravity and a lightweight construction ($< 20\text{Kg}$). It is designed to operate over a range of urban, off-road and industrial terrains including restricted access

areas and traverse both curbs (road/side-walk transition) and stairways. This is typical of the small, man-portable tele-operation platforms available for both EOD and Chemical, Biological, Radiological and Nuclear (CBRN) investigative operations [25]. Video is transmitted from the platform as an analogue PAL video signal at 25 fps using standard VHF video transmission (representing worst case video quality). Transmission noise is present (and noticeable) within the received video for operator tele-operation. Robot drive speed is limited to a maximum of ~ 4 mph (representing the worst case scenario for high frequency vibration). Due to the nature of EOD and CBRN operations such a vehicle is generally deployed within an environment that can be assumed to be visually static.

Figure 1. Generalized six degree of freedom camera motion model

The vehicle vibration can be characterized within six degrees of freedom of the on-board camera mounted on the telescopic boom arm of the robot (Figure 1). To ascertain the magnitude of the vibration within this 3D space fixed markers were placed on the robot camera mount and manually analyzed for position displacement within an observational (off-vehicle) video sequence of robot drive motion (Figure 2). In Figure 2 we see two examples:- on the left we estimate the translation in the image plane from a frontal drive observation and on the right the rotational tilt due to side-on observation. From the empirical analysis of robot motion under these observation conditions the over a range of terrain and obstacle traversals, using of fixed scene markers within the on-board camera field-of-view, the relative importance of the different camera motions can be identified. In general it was observed that undesired rotational movements of the camera are insignificant compared to the translational movements when the (imperfect) ego-motion of the vehicle is additionally considered. For

typical forward/reverse drive operation of the robot, translation along the axis of the camera can also be ignored (Figure 1). As such, we solely consider two translational degrees of freedom relating to the x and y directional translation of the video image (Figure 1).

Figure 2. Characterization of position displacement using image markers for image plane (left) and tilt rotation (right)

The maximal level of displacement (in y) due to vibration was observed as up-to ± 35 pixels between two consecutive images occurring at a temporal frequency of ~ 10 Hz over the transmitted video sequence. These two characteristics combine to produce an integrated displacement magnitude of ~ 350 pixels/s which must be removed by any video stabilization approach. Displacement in x was observed to be lesser, as is to be generally expected in vehicular vibration characteristics [21]. Notably the magnitude and frequency characteristics of this platform, typical of tele-operation in hazardous environments [25, 28], differs significantly from prior work on this topic [1, 22–24, 28].

4. PROPOSED METHODOLOGY

Our proposed approach conforms to the typical pipeline of motion *estimation - filtering - compensation* with motion estimation between the current video frame, t , and the previous frame, $t - 1$, forming the backbone. However, we deviate somewhat from prior complex motion model approaches [31–35] to employ a matching variant of [44], limited to a 2D $x \leftrightarrow y$ motion model, from which strong motion estimation pre-filtering facilitates the utilization of a simplistic modeling assumption within the bounds of real-time operation. Here we concentrate our computational effort at the filtering stage, facilitating both the use of simple feature matches [18, 35, 44] and high confidence in post-filtered motion vectors, whilst using a computationally inexpensive final estimation model. By contrast the prior use of differential approaches (e.g. [36, 37]) have notable computational complexity, illumination sensitivity and potential aperture problems for the high magnitude/frequency vibration characteristics with which we

are dealing (see Section 3). In addition, explicitly feature driven approaches (e.g. [43]) can suffer in feature-sparse environments or in the presence of significant motion blur. Our requirements direct us to an approach that can be realized in real-time in an unknown, potentially feature-sparse environment with computational complexity that is readily realizable in hardware [4, 44].

We propose the use of a multi-scale image pyramid decomposition approach [4] as a key enabler to the successful use of block-matching [44] within a real-time context for initial inter-frame Local Motion Vector (LMV) estimation. Furthermore we propose the use of a novel set of outlier rejection criteria as a robust filtering methodology, enabling the effective use of LMVs obtained in this way, as a computationally efficient route to Global Motion Vector (GMV) estimation and subsequent motion compensation (i.e. stabilization).

4.1 Local Motion Vector Estimation

The current video frame, t , and the previous frame, $t - 1$, are first formulated into a multi-scale image pyramid using the technique of [4]. An image pyramid of n levels is constructed, for $i = \{0 \dots n - 1\}$, with each subsequent level, L_i , constructed from a down-sampled version of the original image using a down-sampling factor of two. This approach operates on the standard gray-scale transformation of the colour RGB (YCrCb) video frame transmitted from the mobile platform.

At each pyramid level a Laplacian of Gaussian (LoG) convolution filter is applied to each frame to extract the second spatial derivative of the image with reduced inter-frame noise [45]. Following the technique outlined in [3] we decompose the real-valued LoG result into two spaces, L_i^+ and L_i^- , representing the magnitude of the positive and negative LoG responses respectively at level L_i . These spaces are secondarily thresholded to remove insignificant components of the responses corresponding to high frequency scene noise where thresholds, $\tau_+ = \tau_- = \frac{1}{25}$, following [3]. Response magnitudes above these minimal thresholds are maintained, forming the L_i^+ and L_i^- components for use in block-matching based LMV estimation. This concept is illustrated in Figure 3 for clarity.

We base our block matching on N blocks spatially distributed uniformly across the image component where N is parametrized as an even number, $N > 4$ extending the concept of [3]. These are defined upon the L_i^+ and L_i^- image components and spatially interpolated over the n levels of the multi-scale image pyramid (using L_0^+ and L_0^- as the pyramid base images)

Figure 3. Example LoG Image Pyramid Construction

as illustrated in Figure 4. The number of blocks utilized, N , represents a trade-off between computational cost (of block matching [3]) and the number of LMV samples available for subsequent GMV estimation.

Figure 4. Block distribution over L_i^+ and L_i^- image pyramid

This topology (Figure 4) is thus utilized as the conduit to the derivation of N LMV estimates (one per block) between video frames t and $t - 1$, over n levels using both the L_i^+ and L_i^- image components for each estimation. Each LMV estimate is computed independently based on a localized search window corresponding to each block position from a coarse (pyramid level $n - 1$) to fine (pyramid level 0) scale. This localized search window is iteratively refined at a given pyramid level, i , based on the identified block match within the corresponding search window at the previous level of coarseness, $i + 1$.

Each block location in video frame $t + 1$ is matched to the previous frame t at each level within the image pyramid based on this defined localized search window. The size of the search window is defined as an area bounded by Δx and Δy in each direction from the corresponding block location in frame t as illustrated in Figure 5. At each position the similarity between block b from frame t and search window position in frame $t - 1$ is calculated using the Number of Non-Matching Points (NNMP) defined as follows [3]:

$$\begin{aligned}
distance_{NNMP}(b) = & \sum_{(x,y) \in b} \{L_{[i]}^+(x,y) \\
& \oplus L_{[t-1]}^+(x+k, y+j)\} \\
& \cup \{L_{[i]}^-(x,y) \oplus L_{[t-1]}^-(x+k, y+j)\}
\end{aligned} \quad (1)$$

for $-\Delta x \leq k \leq \Delta x$, and $-\Delta y \leq j \leq \Delta y$ where (x, y) corresponds iteratively to each pixel location in block area b ($\oplus$ = logical XOR, $\cup$ = logical OR). The position within the search window with the lowest $distance_{NNMP}$ (i.e. maximum similarity) identified in frame $t-1$ is compared with the original position of block b to compute the LMV at level i within the pyramid. This is performed iteratively from level n to 0 in the image pyramid with the starting position for this block matching at level $i-1$ adjusted based on the identified match (highest correlation) at level i .

Figure 5. Localized Frame to Frame Block Matching

This adjustment of matching location as we move from one pyramid level to the next, L_i to L_{i+1} , is performed by essentially moving the starting centre location of the search window, $(cx_{b_{L_i}}, cy_{b_{L_i}})$, at subsequent level L_{i+1} to the position of the identified maximum correlation at L_i for the corresponding block b :

$$(cx_{b_{L_{i+1}}}, cy_{b_{L_{i+1}}}) = \arg_{(x,y)} \min(distance_{NNMP}(b_{L_i})) \quad (2)$$

This adjustment essentially uses the LMV estimate from the previous level in the pyramid as the starting position for the search window in the subsequent level (additionally scaling by the pyramid scale factor as we change levels). As the LMV estimates are transmitted between the pyramid levels this allows us to keep the search window size, $(\Delta x, \Delta y)$, constant over all pyramid levels despite the change in image scale (as large LMV estimations are now detected as a series of smaller LMV estimates over multiple pyramid levels). This reduction in LMV estimation search space for our block matching approach facilitates the use of a larger number of blocks than prior work (extending [3]) whilst maintaining a low

computational overhead that supports real-time stabilization performance.

Although we base our LMV estimates purely upon rectangular block matching recent work has also successfully considered circular localized region (circle “blocks”) within this context [46]. This is left as an area for future work within the context of the vibration characteristics and application we are considering.

An example set of LMV estimates using our proposed approach are visualized for two frame examples in Figure 6 where we see statistically consistent (green - Figure 6 left/right) and inconsistent (red - Figure 6 right) vector estimations. This illustrates the need for robust statistical outlier filtering of the LMV field prior to estimation of global motion (i.e. GMV).

Figure 6. Example unfiltered LMV output

4.2 LMV Outlier Rejection

Prior to our consideration of outlier rejection via filtering of the LMV field determined using the previously outlined methodology we first identify the primary two conditions under which they occur:- a) region uniformity and b) block aperture limitation. Image regions which are highly uniform in nature and essentially featureless are not well suited to a block matching based LMV estimation approach. In these regions, the lack of strong image features results in the prevalence of noise in the block matching process. Assuming a random distribution of noise in the image [45] this effectively results in randomized inter-frame matching for such feature sparse image regions. By contrast the aperture problem effects image regions where periodic features/textures are present such that a consistent locally maximal block match cannot be found due to the prevalence of several high correlation results in the block matching process - any difference between which is only attributable to image noise. An example of this is illustrated in Figure 7 where we see an ambiguity between potential local block matches in frame $t-1$ (left, green) to spatially corresponding regions in frame t (right, red) due to repeating scene features/patterns.

Figure 7. Block aperture limitation effects locally consistent matching in the presence of repeating scene features/patterns.

In order to counter the effect of these two negative feature occurrence events on LMV field calculation we introduce two stages of statistical filtering. Firstly, to detect regions of uniformity co-occurring with our block samples of the image frame we use the entropy information metric, E_b , of the image region corresponding to each block, b , as follows:

$$\begin{aligned}
 E_b &= E_b^+ + E_b^- \\
 &= - \sum_{k=0}^1 p_{(k|L_0^+ \cap b)} \log p_{(k|L_0^+ \cap b)} \\
 &\quad - \sum_{k=0}^1 p_{(k|L_0^- \cap b)} \log p_{(k|L_0^- \cap b)}
 \end{aligned} \quad (3)$$

where $p_{(k|L_0^+ \cap b)}$ is the probability of occurrence of pixel value k in the pixel distribution of the image region defined by the spatial intersection of image component L_0^+ (i.e. base level of the earlier image pyramid) and the image region defined by block b (i.e. $L_0^+ \cap b$). By similarity for image component L_0^- also. For clarity, this follows the standard definition of image entropy, E [45], and uses a regular normalized histogram of the pixels values in the image region defined by the block position in the image components L_0^+ and L_0^- derived earlier. Entropy essentially defines the randomness or spread of data distribution as illustrated in Figure 8.

Figure 8 illustrates the relationship between LMV outliers and occurrences of region uniformity / block aperture (Figure 8A) and the corresponding entropy metric for the block samples (Figure 8B). A simple entropy threshold, τ_e , can be defined to isolated occurrences of region uniformity (as illustrated - Figure 8B, red hatched entropy values for $\tau_e = 0.1$).

A separate approach is required for occurrences of ambiguous block matches due to block aperture limitation (see examples in Figure 8A). Generally these

Figure 8. Example entropy metric calculations for frame blocks

occurrences are attributable to single orientation linear scene features that locally dominate a given block location (Figure 8A). The problem is further complicated by the additional cases of parallel edges and periodic or locally symmetric images features/textures which move the derivation of a generalized solution away from the arena of simple edge feature detection [45]. By visualizing the normalized blocking matching correlation results, Eqn. 1, over a localized search window for block matching we can readily appreciate that block matching in such cases does not yield a unique (and robust) frame to frame block match (Figure 9). This contrasts strongly with the example of Figure 10 where we have an unambiguous non-aperture block matching case.

Discarding the use of a simple mean and standard deviation filter to identify blocks with a narrow statistical cluster of correlations (which empirically were found to give poor results) we centre on an approach considering the distribution of correlation distances both directly (in terms of relative value) and in terms of spatial distribution within the block. We propose a threshold based measure whereby a threshold, $\Delta distance_{NNMP}$, defines a similarity measure indicative of an aperture effected block correlation result. Furthermore we consider the spatial offset, $(\Delta x_{NNMP}, \Delta y_{NNMP})$, of all the locations found within the block that have a correlation dis-

Figure 9. Visualization of block matching correlation distance for block aperture examples from Figure 8

Figure 10. Visualization of block matching correlation distance for non-aperture example from Figure 8

tance within $\Delta distance_{NNMP}$ of the minimum correlation value present. This offset is defined from the location of the minimum value present within the block (see Figure 11). If either of these offset values, $(\Delta x_{NNMP}, \Delta y_{NNMP})$, are greater than a given allowed threshold, $\tau_{\Delta xy}$, we consider the distribution of minimal correlation values to be significantly dispersed within the block correlation result. The LMV estimate from this block is then considered as ambiguous and discarded from consideration in subsequent GMV estimation.

This block aperture limitation filtering is performed upon the final block correlation results derived from L_0 , the image pyramid base. Empirically we use $\Delta distance_{NNMP} = 35$ ($\approx 14\%$) for the 8-bit integer

range normalized correlation distances as shown in Figures 9 - 11 and $\tau_{\Delta xy} = 3$. Typically this combination of entropy based (region uniformity) and correlation distribution (block aperture) filtering will remove most but not all LMV outliers.

Figure 11. Spatial distribution measure of block matching correlation

Finally two further stages of filtering consider the remaining distribution of LMV as polar co-ordinates and implement a methodology of magnitude and angle filtering in polar space. An example of the post {uniformity/aperture} LMVs (with manually identified outliers) is shown in Figure 12 in both original (left) and polar co-ordinate form (right, with outliers in red). Filtering based on the polar magnitude is simply based on the (statistically robust) median of the remaining LMV field with any LMV estimate with an absolute difference (from median) in magnitude over a given threshold, τ_{mag} , discarded.

Polar angle filtering based upon a comparative voting scheme whereby every possible pair of angles within the polar representation of the LMV field is compared. Pairs with an angular distance over a given threshold, $\tau_{orientation}$, receive a vote. The LMV corresponding to the polar angle with the greatest number of votes is discarded and this voting procedure is repeated iteratively until no remaining LMV are eliminated in a given iteration.

Empirically we use $\tau_{mag} = 4$ pixels and $\tau_{orientation} = 15^\circ$ in our current formulation (which are both defined relative to image resolution)

4.3 Global Motion Vector Estimation

Based on our four outlier detection filtering approaches over the original LMV field the most common way to proceed in GMV estimation is based on a least squares

Figure 12. LMV field (left) plotted in polar co-ordinates (right)

formulation. Least squares approaches have shown to be computationally expensive for full-frame rate stabilization [19, 37, 44].

Instead we opt for an overly simplistic approach to GMV estimation in comparison to those employed in prior work [19, 31–35, 37, 44]. By contrast, here we rely upon the robustness of our LMV outlier rejection and boldly opt for the computationally efficient, although acknowledge-ably outlier intolerant, approach of deriving the GMV as the simple vector mean of the post filtered LMV field.

We employ a simple heuristic in GMV estimation to prevent GMV estimation (and subsequent motion compensation) if the number of LMV remaining post outlier rejection filtering is less than three. This prevents overly simplistic GMV estimation based on limited LMV samples and stabilization of this frame is foregone, within the video stream, in this case. The impact of this, on 25fps (25Hz) video, is not apparent in the results (see Figures 17 - 26).

4.4 Motion Selection and Compensation

The global motion estimate at this stage (i.e. GMV) contains both the ego-motion of the platform itself (in response to tele-operation control demands) and unstable vibrational motion due to the terrain. For successful video stabilization we desire to preserve the ego-motion within the resulting video whilst removing the effect of the unstable motion.

These two motion components can be distinguished based on their inherent frequency characteristics. In general the vibrational component of the global motion has a frequency, f_v , in the range $\tau_l < f_v < \tau_u$ between upper and lower frequency thresholds τ_u and τ_l with the frequency of the ego-motion, f_e , occurring below this range $0 < f_e < \tau_l$. For the short wheelbase platform considered here, empirically we establish that $\tau_l \approx 4Hz$ and $\tau_u \approx 12Hz$.

Before moving forward, we additionally define two concepts we will work with:- the *Relative Global Motion* (RGM) and the *Absolute (or Accumulative) Global Motion* (AGM). The RGM corresponds to the global motion vectors estimated between each consecutive frames while the AGM represents the accumulation of these global motion vectors with respect to the first frame of the video sequence (illustrated in Figure 13). We consider operations on the RGM and AGM independently on the horizontal (x -axis) and vertical (y -axis) components. For simplicity the following explanation will only consider the vertical camera motion (i.e. y -axis components of the GMV, RGM and AGM). The same formulation is repeated on the horizontal camera motion (x -axis) to produce the final stabilized video output.

Figure 13. Exemplar RGM and AGM plotted over n video frames (vertical component only)

Motion compensation filtering is based on an RGM/AGM motion signal (Figure 13) at given time (frame) t where the previous $t-1$ time (frame) samples are already known. This formulation makes the application of a traditional low-pass filtering approach in the frequency domain [45] not readily practical. To counter this we use a real-space based approach operating as a temporally weighted filter over the RGM samples directly. A weighting function, $w_i = f(t, l)$, defines a relative weight to a given RGM sample, $t-l$, based on its relative temporal offset, l , from t . This function is defined over a given discrete offset window (or range of samples), $l = [0..l_{max}]$, such that $f(t, l) > 0$ for $l \leq l_{max}$ and $f(t, l) = 0$ for $l > l_{max}$. Subsequently, we derive a filtered GMV as follows:

$$GMV'[t] = \sum_{l=0}^{l_{max}} w_l GMV[t-l] \quad (4)$$

where $GMV[t]$ is the Global Motion Vector estimation for frame t . This essentially provides a temporally weighted version of the raw GMV estimation at frame t . Empirically we use a weighting function, $f(t, l)$, that returns linearly distributed weights normalized to the range $[0 \rightarrow 1]$ and a window size, $l_{max} = 7$ (given unstable motion in frequency range $4-12Hz$ and a video

frame sample rate of 25fps). This smoothed version of $GMV[t]$, itself a measure of the RGM at frame t , is inverted to provide the compensation vector $CV[t]$ such that $CV[t] = -GMV'[t]$. This is repeated for both the horizontal (x -axis) and vertical (y -axis) GMV components.

Figure 14. Influence of window size parameter on motion filtering (vertical component)

From our results (Section 5) we demonstrate that despite its simplicity this basic motion selection approach is able to extract the undesirable part of the RGM. Furthermore, simple subtraction of this (pre-filtered) component applied to each frame removes the undesired motion component of the original video source. In Figure 14 we illustrate the influence of the window parameter, l_{max} , upon the filtering of the vertical component of the RGM. In this figure (Fig. 14) we see this simple compensation approach, itself resting upon the robust filtering of LMV selection, removing varying levels of vibration related motion whilst maintaining the overall (signal) characteristics of the ego-motion component of the GMV. As is evident from the plot of Figure 14 a frame correspondence delay of two frames (80ms for 25 fps) is introduced into the resulting filtered (stabilized) video output. This delay is inherent in the frame to frame correspondence approach used - i.e. the GMV calculated between frames $t - 1$ and t is output (after processing post arrival of frame t) as frame $t + 1$ arrives

for processing. This delay is, however, well within the accepted bounds for effective tele-operation [25, 26, 28].

4.5 Panning Motion Detection

The only remaining issue is that of intentional camera panning motion, by tele-operated command, in the horizontal or vertical axis. Unchecked, our counter-vibration motion compensation can introduce a more significant perceptual delay in the video displayed to the tele-operator in this case. This is illustrated in Figure 15A where we see a four frame (160ms) delay introduced during panning motion of the camera (robot motion otherwise stationary). To counter this we use explicit panning detection in the form of a consistency check against the direction component of the GMV in either axis. If it is consistent for u frames we assume a commanded panning motion is taking place and deactivate our motion compensation on the video output for v subsequent frames (still re-checking motion consistency at each frame). Whilst this still introduces an initial panning detection delay of u frames (i.e. 40u milliseconds for 25 fps) it allows the immediate cessation of this delay as motion compensation is deactivated (see Figure 15B). This approach of deactivating stabilization for explicit panning motions facilitates immediate “catchup” of the tele-operation video (via frame skip) to show the relative original and smoothed AGM signals of Figure 15B. Although the perceptive delay within the tele-operation video is limited, operationally panning motion occurs in a number of situations including vehicle turning, manipulation of the on-board telescopic arm and stairway/step traversal.

4.6 Final Compensated Frame Rendering

The final stage of motion compensation deals with the issue of relative image size post application of $CV[t]$ to the frame. As a given frame, t , is transformed by the correction vector, $CV[t]$, it introduces irregularly occurring cropping on varying borders of the frame as it is translated in a given direction towards the opposing side of the image (e.g. Figure 16A). This is compensated for by identifying the largest component (x or y) spacing of irregular cropping, applying a uniform cropping operation to the image based on this component (Figure 16B) and re-scaling the image (linear interpolation) based on an anchored central position (i.e. re-scaled outwards from centre, Figure 16B).

5. RESULTS

We evaluate our proposed approach over varying urban terrain, typical for tele-operational use of a robot of

Figure 16. Irregular frame cropping (A, upper) compensated with crop/rescale (B, lower)

Figure 15. Illustrative horizontal panning motion AGM with (B) and without (A) initial panning motion detection

this type/size, including instances of obstacle traversal (e.g. step/kerb) and additionally evaluate the real-time performance of the algorithm under varying parameter conditions.

5.1 Varying Terrain Conditions

Firstly we operate the robot in the traversal of poor quality, uneven tarmac (road surface) including a “pot-hole” of approximate diameter 15cm and 5cm depth (significant given the size/dimensions of the robot platform). The resulting, un-stabilized, video from this experiment is shown in Figure 17 where we illustrate the magnitude and direction of the vibrational motion effects present over a subset of eight frames from the

original sequence. In Figure 17 we maintain the cross-hairs (red) in a fixed position and translate the image based on the compensation vector, *GMV*, estimated using the methodology outlined in Section 3 without applying the final motion compensation element of Section 4.4. Using the fixed position cross-hairs as a reference and noting the oscillation of scene features present in each frame, such as the car (Figure 17), above and below these cross-hairs (despite constant forwards ego-motion) we can appreciate the vibration effects present in this test sequence.

In Figure 18 we show the corresponding AGM displacement plot of this sequence before (black) and after (red) application of our stabilization approach. We generally observe a small delay (3-4 frames) between the two AGM plots but the main trend of the ego-motion is conserved. Notably this delay is not the absolute delay (identified as 2 frames in Section 4.4) but the notional delay introduced due to the effects of the smoothing implicitly applied to the AGM by the temporal filtering approach employed. In the early part of the sequence many oscillations with a high frequency have been completely removed (Figure 18 left). The highlighted part of the signal, which corresponds to the video frames of Figure 17, is perturbed by very high level of displacements. We can notice that the maximal AGM (just within the highlighted section, Figure 18) reaches 30 pixels in magnitude between two frames and many displacements of ~ 20 pixels are detected and attenuated on the middle section of the AGM plot (Figure 18). Although it is interesting to note that vibration due to such a consequent obstacle traversal are not completely eliminated the majority of high frequency video oscillation is removed. It is important to conserve this kind of situational awareness information for conveyance to

Figure 17. Sub-sampled video to illustrate vibration present in traversal of uneven road surface

the operator (i.e. obstacle traversal) whilst removing the high frequency video oscillations that make tele-operation difficult due to visual fatigue.

Figure 18. AGM plot corresponding to sequence of Figure 17

In Figures 19 to 21 we illustrate our approach over three video sequences from the tele-operated platform in use (Section 3). In these examples (Figures 19 - 21) we show a sequence of extracted frames where the left side is the original (un-stabilized) video and the right is the motion compensated approach (separated by the vertical cross-hair). In contrast to examination of the effect of motion on the AGM plot (e.g. Figure 18) we seek to illustrate the actual improved perceptual effect our approach has in terms of object stabilization within a given sequence. Whilst highly subjective, yet within the limitations of presenting this material in printed form, we can visually assess the effects of stabilization by comparing the relative motion (to the fixed cross-hairs) of objects in the left and right portions of the video frames (Figure 19 - 21). In Figure 19 if we compare the motion of the blue car (left) to that of the building windows (right) during forward motion we can see a contrast between erratic motion (left) and consistent change/enlargement in perspective (right). Similarly

in Figure 20 (frames A-D) we can see a contrast between the fluctuations of the vertical position of the tops of the trees (left) and the more consistent vertical position of the fence (right). Figure 20 (frames E-G) show a less obvious stabilization effect but show both left-right artefacts due to correction (e.g. gray building / road markings) and a consistent vertical position of the fence and white van object under stabilization (right). In Figure 21 we see the blue fence object (right) consistently advancing to the horizontal centre line (and crossing it in Frames G/H) due to the perspective effect of forward motion whilst in Figure 21 (left) we see more erratic behaviour of the building windows relative to the fixed cross-hairs. Appreciation of all sequences is notably improved in a video form and the reader is directed to <http://www.cranfield.ac.uk/~toby.breckon/demos/stabilization/> for further qualitative evaluation.

5.2 Re-activity Conservation

We consider at the conservation of re-activity (i.e. responsiveness) within the stabilized video with respect to two aspects - a) active re-activity (i.e. responsiveness to operator commands) and b) passive re-activity (i.e. responsiveness to terrain features). Given that Section 4.4 outlines a fixed 2 frame delay in the stabilized video output (i.e. 80ms for 25fps) our interest here is focused on the active/passive responsiveness due to the use of panning detection that deactivates stabilization in response to consistent camera panning motion.

Firstly we consider active re-activity, with respect to panning detection. In Figure 22 where we see the original and stabilized AGM plots for “on the spot” rotation

Figure 19. Illustrative frame extracts from building approach sequence (1)

Figure 20. Illustrative frame extracts from building approach sequence (2)

of the vehicle at two different speeds. These speeds (measured in pixels/s. in terms of average image flow in the scene) represent a low speed turn (50 pixels/s., Figure 22a) and a high speed turn (200 pixels/s., Figure 22b). From Figure 22a we see an initial small detection delay of 4 frames (highlighted) until panning detection is activated and the original and filtered AGM plots fall into co-incidence with stabilization effectively disabled until panning motion ceases (as expected). In Figure 22b we can again see that at the beginning of each panning motion (denoted as A, C and E in Figure 22b) a small delay is introduced by panning detection. Furthermore we notice a similar small re-activity delay at the end of a panning motion (denoted as B and D in Figure 22b) where the motion is not correctly compensated post-panning. In all cases the re-activity de-

lay is minimal and limited to the stated 4 frame delay of panning detection incurred due to the design of the proposed approach (i.e. 160ms with 25fps video). This delay, occurring only post-panning motion, is tolerable for operational use in these isolated occurrences.

Secondly we consider passive reactivity which is evaluated as the vehicle traverses a up-hill slope (see Figure 23a - photo inset, slope highlighted) that causes a natural upward panning motion of the camera. We consider the effect of the panning detection approach within the stabilization of the video under this type of passive panning motion (due to obstacle traversal).

In Figure 23a we see the effect of the activation of automatic panning detection in the AGM (sections denoted A, B and C) where the video has either not been

Figure 21. Illustrative frame extracts from building approach sequence (3)

Figure 22. AGM Active Reactivity Example

stabilized due to the deactivation of stabilization by detection of a panning motion (A/B) or incorrectly stabilized (filtered) (B/C) during this obstacle traversal. If we deactivate the panning detection itself we can see

the quality of the stabilized AGM is greatly improved (Figure 23b). However, a delay of ~ 3 -4 frames (~ 160 ms for 25fps video) is introduced over the panning motion (see Figure 23b - AGM gradients). Furthermore we can observe this within the visualization of Figure 24 where we see a sub-sampled split video visualization presented in the same manner as the earlier examples in Figures 19 - 21 (left of frame = original, right of frame = stabilized). Through the sequence Figure 24, frames A-H, we can see as the vehicle begins the ascent, the stabilized version (right) tends to maintain the original horizontal position stable (notably frames A-C, F-G).

Overall whilst the change in re-activity from the *de-facto* 2 frame (80ms at 25fps) delay of the proposed approach to up to 4 frames (160ms at 25fps) is notable for the use of automatic panning detection, the impact of this is negligible within an operating scenario where active/passive panning is limited. The impact of automatic panning detection on the overall stabilization of the video during potential obstacle traversal is also noted. Given the fixed overhead of panning detection, and its impact on re-activity/stabilization, the use of this technique is left for consideration on a per scenario basis and as an area for future work.

5.3 Real-time Performance

We further explore the real-time performance of the proposed approach with respect to the various parameter choices outlined in Section 3. The parameter choices of the number of pyramid levels n , block size and number of blocks N and search window size $\Delta x \Delta y$ are explored with respect to their impact on performance. Performance is measured as the mean effective frame-rate (frames per second, fps) throughput over a set 2000

Figure 23. AGM Passive Reactivity Example

frame video sequence comprising of varying motion of the platform over varying environmental conditions. A base configuration of for pyramid levels ($n = 4$), search window size of 3×4 ($\Delta x \times \Delta y = 3 \times 4$) pixels, block size 30×40 pixels and a block layout of 5×5 ($N = 25$) blocks. Each of these parameters is then modified independently to evaluate the influence on performance.

Figure 25. Effect of varying pyramid levels on frame-rate

In Figure 25 we can see the increase in frame-rate performance as we increase the number of pyramid levels. This is attributable to the reduction in the computa-

tional cost of block matching as a result of increased multi-scale processing. However, as shown in Figure 26, we see a gradual rise in the percentage of frames discarded (due to insufficient inlier block matches) and yet a steady fall in the mean number of outlier blocks per (non-discarded) frame for pyramid levels, $n = 1..4$. Essentially this trend supports a conclusion that as the number of pyramid levels is increased estimation is based on fewer, higher quality frame motion estimates. The notable, and contrasting, sharp increase in both for $n = 5$ is attributable to the interaction of the down-scaling effect of five pyramid levels and the interaction of the block size and search window parameters for a 5×5 block layout. At this level we arrive at overlapping block search windows in an image with significantly sparse texture detail remaining resulting in pseudo-random block matches in many instances. Such matches are successfully identified as outliers using the approaches outlined in Section 3.

By contrast, in Figures 27 and 28 we confirm the somewhat predictable result that effective frame-rate is reduced as either of block size or the number of match-

Figure 24. Illustrative frame extracts from slope obstacle traversal

Figure 26. Effect of varying pyramid levels on outlier block number

ing blocks is increased. Block size (for an $X \times X$ block) varies quadratically as one would expect with this $O(X^2)$ relationship whilst the number of blocks present varies linearly as expected.

Figure 27. Effect of varying block size on frame-rate

The effect on performance of varying the search win-

Figure 28. Effect of varying number of blocks on frame-rate

dow size, $\Delta x \Delta y$, is shown in Figure 29. Here we see a subtler relationship with a “stepped” linear trend. Whilst the linear relationship is to be expected, the “stepped” nature is attributable to the integer rounding of the re-partitioning of the search window size in the lower levels of the pyramid such that (for example) $\Delta x \times \Delta y = 100$ generates the same search window sizes for pyramid levels $i > 2$. The “step” characteristic represents a increase in the matching cost of these lower levels when round accumulation reaches the next whole pixel width division.

Overall from Figures 25 - 29 and our analysis of Sections 5.1 and 5.2 we empirically derive a set of parameters that achieve our benchmark performance of real-time stabilization at 25fps using a $N = 25(5 \times 5)$ block arrangement of 30×35 pixel dimension blocks using a $n = 4$ level pyramid. This is based on 352×288 resolution video image operating on a 1.8GHz (single-core) CPU. As discussed in Section 5.2 a processing delay of 80ms (or in general $2 \times (frame\ period)$) is inherent in the proposed methodology independent of available computational processing power.

Figure 29. Effect of varying matching window size on frame-rate

6. CONCLUSIONS

Overall we present a real-time video stabilization approach for use with the vibration characterizes of a short-wheel base, non-suspension tele-operated vehicle. By adopting a novel four-stage filtering methodology for the identification of motion vector (LMV) rejection we facilitate use of both an efficient block matching approach [3] and a simplistic, thus efficient, global motion estimation model. Furthermore we extend the block matching methodology of [3] over a multi-scale pyramid [4] to achieve further computational saving. Results are presented over a range of environments with evaluation both by qualitative visualization and quantitative assessment of both original and filtered motion accumulation. Stabilization is achieved with a 2-frame (i.e. 80ms at 25fps) visualization delay rising to isolated 4-frame (i.e. 160ms at 25fps) occurrences with the activation of the proposed pan motion detection approach within the overall methodology.

Future work will investigate the use of dynamic block placement and sizing [46] based on real-time scene content analysis and improved automatic panning motion detection.

The authors gratefully acknowledge the support of QinetiQ Ltd (UK) in this research activity.

REFERENCES

1. R. Miyauchi, N. Shiroma, and F. Matsuno, "Compact image stabilization system using camera posture information," *J. of Field Robotics*, vol. 25, no. 4-5, pp. 268–283, 2008.
2. A. Chenebert, T. Breckon, and A. Gaszczak, "A non-temporal texture driven approach to real-time fire detection," in *Proc. International Conference on Image Processing*, pp. 1781–1784, IEEE, September 2011.
3. K. Nam-Joon, L. Hyuk-Jae, , and L. Jae-Beom, "Probabilistic global motion estimation based on laplacian two-bit planematching for fast digital image stabilization," *EURASIP J. on Advances in Signal Processing*, pp. 1–10, 2008.
4. P. Burt and E. Adelson, "The laplacian pyramid as a compact image code," *IEEE Trans. Communications*, vol. 31, pp. 532 – 540, apr 1983.
5. T. Breckon, S. Barnes, M. Eichner, and K. Wahren, "Autonomous real-time vehicle detection from a medium-level UAV," in *Proc. 24th International Conference on Unmanned Air Vehicle Systems*, pp. 29.1–29.9, March 2009.
6. A. Gaszczak, T. Breckon, and J. Han, "Real-time people and vehicle detection from UAV imagery," in *Proc. SPIE Conference Intelligent Robots and Computer Vision XXVIII: Algorithms and Techniques*, vol. 7878, 2011.
7. M. Magnabosco and T. Breckon, "Cross-spectral visual Simultaneous Localization And Mapping (SLAM) with sensor handover," *Robotics and Autonomous Systems*, vol. 63, pp. 195–208, February 2013.
8. M. L. Eichner and T. Breckon, "Real-time video analysis for vehicle lights detection using temporal information," in *Proc. 4th European Conference on Visual Media Production*, pp. I–9, IET, November 2007.
9. M. L. Eichner and T. Breckon, "Integrated speed limit detection and recognition from real-time video," in *Proc. IEEE Intelligent Vehicles Symposium*, pp. 626–631, IEEE, June 2008.
10. I. Katramados, S. Crumpler, and T. Breckon, "Real-time traversable surface detection by colour space fusion and temporal analysis," in *Proc. International Conference on Computer Vision Systems*, vol. 5815 of *Lecture Notes in Computer Science*, pp. 265–274, Springer, 2009.
11. A. Kheyrollahi and T. Breckon, "Automatic real-time road marking recognition using a feature driven approach," *Machine Vision and Applications*, vol. 23, no. 1, pp. 123–133, 2012.
12. I. Tang and T. Breckon, "Automatic road environment classification," *IEEE Trans. on Intelligent Transportation Systems*, vol. 12, pp. 476–484, June 2011.
13. F. Mroz and T. Breckon, "An empirical comparison of real-time dense stereo approaches for use in the automotive environment," *EURASIP Journal on Image and Video Processing*, vol. 2012, no. 13, pp. 1–19, 2012.
14. O. Hamilton, T. Breckon, X. Bai, and S. Kamata, "A foreground object based quantitative assessment of dense stereo approaches for use in automotive environments," in *Proc. International Conference on Image Processing*, IEEE, September 2013. to appear.
15. J. Sokalski, T. Breckon, and I. Cowling, "Automatic salient object detection in UAV imagery," in *Proc. 25th International Conference on Unmanned Air Vehicle Systems*, pp. 11.1–11.12, April 2010.
16. I. Katramados and T. Breckon, "Real-time visual saliency by division of gaussians," in *Proc. International Conference on Image Processing*, pp. 1741–1744, IEEE, September 2011.

17. T. Breckon, J. Han, and J. Richardson, "Consistency in multi-modal automated target detection using temporally filtered reporting," in *Proc. SPIE Electro-Optical Remote Sensing, Photonic Technologies, and Applications VI*, vol. 8542, pp. 23:1–23:12, November 2012.
18. K. Ratakonda, "Real-time digital video stabilization for multi-media applications," in *Proc. Int. Symp. Circuits and Systems, 1998*, pp. 69–72, 1998.
19. H. Chang, S. Lai, and K. Lu, "A robust and efficient video stabilization algorithm," in *Proc. Int. Conf. Multimedia and Expo*, pp. 29–32, 2004.
20. C. Wang, J. Kim, K. Byun, J. Ni, and S. Ko, "Robust digital image stabilization using the kalman filter," *IEEE Trans. Consumer Electronics*, vol. 55, no. 1, pp. 6–14, 2009.
21. L. Bombini, P. Cerri, P. Grisleri, S. Scaffardi, and P. Zani, "An evaluation of monocular image stabilization algorithms for automotive applications," in *Proc. Intelligent Transportation Systems Conf.*, pp. 1562–1567, 2006.
22. N. Shiroma, J. Kobayashi, and E. Oyama, "Compact image stabilization system for small-sized humanoid," in *Int. Conf. Robotics and Biomimetics*, pp. 149–154, 2009.
23. Y. Choi, T. Kang, and S. Lee, "Development of image stabilization system using extended kalman filter for a mobile robot," *Advances in Swarm Intelligence*, vol. 6145, pp. 675–682, 2010.
24. M. Smith, A. Boxerbaum, G. Peterson, and R. Quinn, "Electronic image stabilization using optical flow with inertial fusion," in *Proc. Intelligent Robots and Systems*, pp. 1146–1153, 2010.
25. T. Fong and C. Thorpe, "Vehicle teleoperation interfaces," *Autonomous robots*, vol. 11, no. 1, pp. 9–18, 2001.
26. P. Hokayem and M. Spong, "Bilateral teleoperation: An historical survey," *Automatica*, vol. 42, no. 12, pp. 2035–2057, 2006.
27. C. Morimoto and R. Chellappa, "Evaluation of image stabilization algorithms," in *Proc. Int. Conf. Acoustics, Speech and Signal Processing*, pp. 2789–2792, 1998.
28. A. Amanatiadis, I. Andreadis, A. Gasteratos, and N. Kyriakoulis, "A rotational and translational image stabilization system for remotely operated robots," in *Proc. Int. Workshop Imaging Systems and Techniques*, pp. 1–5, may 2007.
29. A. Broggi, C. Caraffi, P. Porta, and P. Zani, "The single frame stereo vision system for reliable obstacle detection used during the 2005 darpa grand challenge on terramax," in *Proc. Intelligent Transportation Systems Conf.*, pp. 745–752, sept. 2006.
30. A. Broggi, P. Grisleri, T. Graf, and M. Meinecke, "A software video stabilization system for automotive oriented applications," in *Proc. Vehicular Technology Conf.*, pp. 2760–2764, 2005.
31. C. Morimoto and R. Chellappa, "Fast electronic digital image stabilization for off-road navigation," *Real-Time Imaging*, vol. 2, pp. 285–296, October 1996.
32. M. Hansen, P. Anandan, K. Dana, G. van der Wal, and P. Burt, "Real-time scene stabilization and mosaic construction," in *Proc. Applications of Computer Vision*, pp. 54–62, dec 1994.
33. Y. Yeh, H. Chiang, and S. Wang, "Digital camcorder image stabilizer based on gray-coded bit-plane block matching," *Optical Engineering*, vol. 40, no. 10, pp. 2172–2178, 2001.
34. J. Jin, Z. Zhu, and G. Xu, "A stable vision system for moving vehicles," *IEEE Trans. Intelligent Transportation Systems*, vol. 1, pp. 32–39, mar 2000.
35. Y. Liang, H. Tyan, S. Chang, H.-Y. Liao, and S. Chen, "Video stabilization for a camcorder mounted on a moving vehicle," *IEEE Trans. Vehicular Technology*, vol. 53, pp. 1636–1648, nov. 2004.
36. Z. Duric and A. Rosenfeld, "Image sequence stabilization in real time," *Real-Time Imaging*, pp. 271–284, 1996.
37. S. Srinivasan and R. Chellappa, "Image stabilization and mosaicking using the overlapped basis optical flow field," in *Proc. Int. Conf. Image Processing*, vol. 3, pp. 356–359 vol.3, oct 1997.
38. J. Derutin, F. Dias, L. Damez, and N. Allezard, "Simd, smp and mimd-dm parallel approaches for real-time 2d image stabilization," in *Proc. Computer Architecture for Machine Perception*, pp. 73–80, 2005.
39. S. Erturk and T. Dennis, "Image sequence stabilisation based on dft filtering," *Proc. Vision, Image and Signal Processing*, pp. 95–102, 2000.
40. A. Litvin, J. Konrad, and W. Karl, "Probabilistic video stabilization using kalman filtering and mosaicing," *Image and Video Communications and Processing*, vol. 5022, no. 1, pp. 663–674, 2003.
41. Z. Juanjuan and G. Baolong, "Electronic image stabilization system based on global feature tracking," *J. of Systems Engineering and Electronics*, vol. 19, no. 2, pp. 228–233, 2008.
42. W. Yang, Z. Zhang, Y. Zhang, X. Lu, J. Li, and Z. Shi, "A real-time gray projection algorithm for electronic image stabilization," in *Int. Symp. on Photoelectronic Detection and Imaging*, vol. 8193, p. 81932G, SPIE, 2011.
43. B. Pinto and P. Anurenjan, "Video stabilization using speeded up robust features," in *Proc. Int. Conf. Communications and Signal Processing*, pp. 527–531, 2011.
44. T. Chiew, J. Chung-How, D. Bull, and C. Nishan Canagarajah, "Rapid block-based global motion estimation and its applications," in *Int. Conf. Consumer Electronics*, pp. 228–229, 2002.
45. C. Solomon and T. Breckon, *Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab*. Wiley-Blackwell, 2010.
46. L. Xu and X. Lin, "Digital image stabilization based on circular block matching," *IEEE Trans. Consumer Electronics*, vol. 52, no. 2, pp. 566–574, 2006.