A Robust Image Alignment Algorithm for Video Stabilization Purposes

Giovanni Puglisi, *Member, IEEE,* and Sebastiano Battiato, *Senior Member, IEEE*

***Abstract*—Today, many people in the world without any (or with little) knowledge about video recording, thanks to the widespread use of mobile devices (personal digital assistants, mobile phones, etc.), take videos. However, the unwanted movements of their hands typically blur and introduce disturbing jerkiness in the recorded sequences. Many video stabilization techniques have been hence developed with different performances but only fast strategies can be implemented on embedded devices. A fundamental issue is the overall robustness with respect to different scene contents (indoor, outdoor, etc.) and conditions (illumination changes, moving objects, etc.). In this paper, we propose a fast and robust image alignment algorithm for video stabilization purposes. Our contribution is twofold: a fast and accurate block-based local motion estimator together with a robust alignment algorithm based on voting. Experimental results confirm the effectiveness of both local and global motion estimators.**

***Index Terms*—Block matching, image alignment, video stabilization.**

I. Introduction

IN THE LAST decade, multimedia devices (personal digital assistants, mobile phones, etc.) have dramatically diffused. Moreover, the increasing of their computational performance combined with a higher storage capability allows them to process large amounts of data. These devices, typically small and thin, usually have video acquisition capability. However, making a stable video is a very challenging task especially when an optical or digital zoom is used. Due to user's hand shaking, the recorded videos suffer from annoying perturbations. The same problem arises in presence of cameras placed on moving supports (e.g., car, airplane) or fixed cameras operating outdoors where the atmospheric conditions (e.g., the wind) and the vibrations produced by passing vehicles make the recorded video unstable.

Video stabilization enables to acquire video sequences without disturbing jerkiness by compensating unwanted camera movements. Video quality is improved and the higher level algorithms present in the device (e.g., segmentation, tracking, recognition) usually work better [1], [2]. Digital video stabilization algorithms, in general, are made up of three stages (see [3] for an alternative scheme): global motion estimation, motion filtering, and image warping [4]. The first stage is devoted to derive the parameters of the transformation occurred between subsequent frames. Translational, similarity, and affine are the most commonly adopted motion models. The second step discriminates intentional motion (e.g., panning) from the unwanted motion (jitter). Typically, motion smoothness is taken into account in the filtering process (jitter is a high frequency signal). Finally, the stabilized image is reconstructed through a proper warping.

Motion estimation is a crucial video stabilization step. Its accuracy heavily influences both motion filtering and image warping. In order to be used in mobile devices, these algorithms have to be very fast, due to the limited resources, both in terms of computing and memory. Another issue relative to the global motion estimation algorithms is their robustness with respect to different scene contents (indoor, outdoor, natural, manmade, etc.) and conditions (illumination changes, moving objects, etc.). Various motion estimation approaches have been used in the video stabilization algorithms. Previous works can be classified in two main categories: direct [5]–[15] and feature-based methods [16]–[20]. The former techniques aim to recover the unknown parameters through global minimization criteria based on direct image information. Some assumptions (e.g., brightness constancy) are typically used as starting point. On the contrary, feature-based approaches first locate a sparse set of reliable features in the image and then recover the motion parameters considering their correspondences.

In this paper, we propose a fast and robust global motion estimation approach for video stabilization purposes. A set of motion vectors is efficiently extracted by a block-matching algorithm. Taking into account the limited block size and the high frame rate, we assume only a translational motion for each block. Such collected information, coming from different spatial locations in the frame, is then used to compute through a voting strategy the global motion vector related to a similarity motion model (two translations, one zoom factor, one rotation). Our block-based local motion estimator has been designed to be as fast as possible maintaining at the same time a high accuracy. This result has been obtained by combining together the exhaustive search strategy with an integral projection-based [21] error function. The novelty of our approach is related to the usage of the gradient of the integral projections (better performance in presence of illumination changes) and the introduction of several optimizations based on mathematical inequalities to reduce the overall complexity. The global motion estimation is then performed

Manuscript received December 15, 2010; revised May 13, 2011; accepted June 24, 2011. Date of publication August 4, 2011; date of current version October 5, 2011. This paper was recommended by Associate Editor Y. Fu.

The authors are with the Department of Mathematics and Computer Science, University of Catania, Catania 95125, Italy (e-mail: puglisi@dmi.unict.it; battiato@dmi.unict.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2011.2162689

by using a filtering step, based on voting (high robustness), followed by a least square estimation (good accuracy). The effectiveness of the proposed approach has been confirmed by a series of tests on video sequences acquired in several challenging conditions: indoor, outdoor, illumination changes, moving objects, periodic patterns, and zoom.

The remainder of this paper is organized as follows. The overall details of the proposed local motion estimator are presented in Section II. Some comparisons with other block-based techniques are also reported. The global motion estimation technique is detailed in Section III. Experimental results are presented in Section IV whereas the computational complexity is reported in Section V. Finally, conclusions are summarized in Section VI.

II. Block-Based Local Motion Estimator

Direct approaches compute frame alignment considering image intensity values. Block-based techniques first divide the image in blocks, typically square, and then search the corresponding one in the next frame. Matching is performed within a search window minimizing a proper error function. Although the exhaustive strategy, full search algorithm (FSA), in which all the possible correspondences are tested, provides good results, it is computational unfeasible. In order to speed up FSA, several strategies have been developed. Some approaches [22]–[24], through simple mathematics inequalities, avoid the computation of the associated error function everywhere without degrading the overall accuracy (i.e., same FSA results with a lower number of operations). Other approaches [25]–[27] do not consider all the possible correspondences. A correspondence is properly found by analyzing just a subset of the overall spatial locations. The computational load is dramatically reduced but, in some cases, these approaches can be misled by local minima, hence, degrading their performances.

The complexity of the FSA is strictly related to the 2-D matching procedure. In order to reduce it to a simple 1-D matching, the integral projections can be taken into account [21], [28], [29]. If two images (or subimages) are similar, their projections even have to be similar. Although, ideally, the reverse is not always true (many 2-D pixel configurations can have the same projections), practically, due to the high frame rate of video applications, consecutive frames are strictly correlated and the reverse can be reasonable considered true. Let I be a $W \times H$ image (or subimage), the integral projections along a generic k column (row) are defined as follows:

$$\mathrm{IP}^V(k) = \sum_{i=1}^{H} I(i,k) \tag{1}$$

$$\mathrm{IP}^H(k) = \sum_{j=1}^{W} I(k,j). \tag{2}$$

The integral projections can be then combined with the exhaustive search (very useful to avoid local minima) obtaining a very efficient block-matching estimator [30]–[32]. However, integral projections can be easily misled by illumination changes, degrading hence the performance of the entire estimator (Section II-B). In order to cope with this problem, the proposed block-based estimator considers the matching of difference curves (Fig. 1) defined as follows:

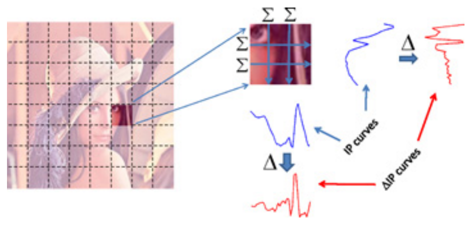

Fig. 1. Example of curves computation. From each block horizontal and vertical cumulative sums are computed (IP curves). ΔIP curves are then obtained by the difference of adjacent IP curve elements.

$$\Delta \mathrm{IP}^d(k) = \mathrm{IP}^d(k+1) - \mathrm{IP}^d(k), \quad d \in \{V, H\}. \tag{3}$$

Just collecting $\Delta\mathrm{IP}^d$ values along columns and rows of each block, the matching needed to estimate local motion vectors could be easily computed. In particular, the sum of absolute differences (SAD) among $\Delta\mathrm{IP}^d$ values of involved blocks, $\mathrm{SAD}_{\Delta\mathrm{IP}}$, has been exploited. This approach, together with some optimizations, permits us to obtain satisfactory results maintaining at the same time low complexity of the overall system. $\mathrm{SAD}_{\Delta\mathrm{IP}}$ can be computed from each pair of $N \times N$ blocks (b_1, b_2) as follows:

$$\mathrm{SAD}_{\Delta\mathrm{IP}}(b_1, b_2) = \sum_{d \in \{V,H\}} \sum_{k=1}^{N-1} |\Delta\mathrm{IP}^d_{b_1}(k) - \Delta\mathrm{IP}^d_{b_2}(k)|. \tag{4}$$

Each individual computation of the error is fairly costly, but considering the entire image an acceptable overall computational cost can be obtained as detailed below. Both $\Delta\mathrm{IP}^d$ curves can be derived in an efficient way as synthetically sketched in Fig. 2.

The underlying idea is similar to the concept of "integral image" [33], an intermediate representation used to rapidly compute 2-D image features. For simplicity, our intermediate representation considers the cumulative sums of row and column image values separately. Details related to the ΔIP computation are provided in the following. First of all, the C^H (C^V) matrix is computed as follows:

$$C^H(i,j) = \sum_{k=1}^{j} I(i,k) \tag{5}$$

where I is the acquired $(W \times H)$ image. Just considering that $\mathrm{SAD}_{\Delta\mathrm{IP}}$ analyzes information coming from pairs of $N \times N$ blocks, it is possible to arrange the cumulative data considering a step of size N obtaining the cumulative sum C^H_N (C^V_N). Starting from C^H (C^V), C^H_N (C^V_N) can be derived as follows:

$$C^H_N(i,j) = C^H(i, j+N) - C^H(i,j). \tag{6}$$

Finally, local differences of adjacent values allow to obtain ΔC^H_N (ΔC^V_N) that comprises all the ΔIP curves needed for block comparisons as follows:

$$\Delta C^H_N(i,j) = C^H_N(i+1, j) - C^H_N(i,j). \tag{7}$$

It is straightforward to verify that the matrices $(\Delta C^H_N, \Delta C^V_N)$ contain all the ΔIP curves. The overall computational cost,

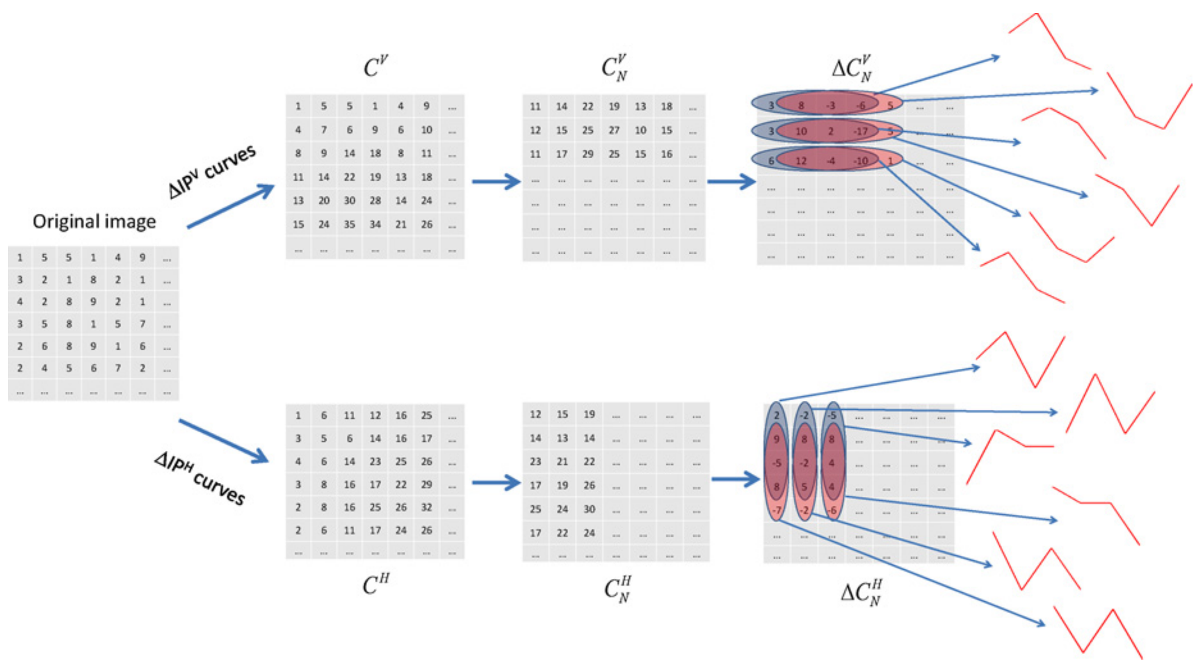

Fig. 2. Example of optimized ΔIP curves computation (for simplicity $N = 4$). All the redundant operations are accurately avoided.

obtained without performing redundant operations, is of $W(H-1)+W(H-N)+(W-1)(H-N+1)$ for $\Delta\mathrm{IP}^V$ curves and $H(W-1)+H(W-N)+(H-1)(W-N+1)$ for $\Delta\mathrm{IP}^H$ curves. The total costs can be hence approximated to 6 WH (H, $W \gg N$): just $6N^2$ additions per block. Thus, considering a search range R, each local motion vector costs (in terms of additions)

$$(2R+1)^2 2(2N-3)+6N^2. \tag{8}$$

A. *Local Motion Estimator Optimization*

Further optimizations can be done to reduce the overall cost just considering some inequalities. Let $\Delta\mathrm{IP}^d_{b_1}$ and $\Delta\mathrm{IP}^d_{b_2}$ ($d \in \{V, H\}$) be two curves of $N-1$ elements. Considering two pairs of adjacent values, we can write the following inequalities:

$$\begin{aligned}&|\Delta\mathrm{IP}^d_{b_1}(k) - \Delta\mathrm{IP}^d_{b_1}(k+1) - (\Delta\mathrm{IP}^d_{b_2}(k) - \Delta\mathrm{IP}^d_{b_2}(k+1))|\\ &= |\Delta\mathrm{IP}^d_{b_1}(k) - \Delta\mathrm{IP}^d_{b_2}(k) + \Delta\mathrm{IP}^d_{b_2}(k+1) - \Delta\mathrm{IP}^d_{b_1}(k+1)|\\ &\le |\Delta\mathrm{IP}^d_{b_1}(k) - \Delta\mathrm{IP}^d_{b_2}(k)| + |\Delta\mathrm{IP}^d_{b_1}(k+1) - \Delta\mathrm{IP}^d_{b_2}(k+1)|.\end{aligned} \tag{9}$$

Considering all the elements of the curves we obtain

$$S^d \quad \le \quad \mathrm{SAD}^d_{\Delta\mathrm{IP}}(b_1, b_2) \tag{10}$$

where

$$\begin{aligned} S^d \quad = \quad & \sum_{k=1}^{N/2} |\Delta\mathrm{IP}^d_{b_1}(2k-1) - \Delta\mathrm{IP}^d_{b_1}(2k) - \\ & (\Delta\mathrm{IP}^d_{b_2}(2k-1) - \Delta\mathrm{IP}^d_{b_2}(2k))|. \end{aligned} \tag{11}$$

Cumulating such results for both vertical and horizontal curves from (10) we have

$$\mathrm{SAD}_{\Delta\mathrm{IP}} \ge S^V + S^H. \tag{12}$$

This result can be used to further reduce the local motion vector computation avoiding some error function evaluations during the block-matching phase. Starting from a generic block b, we are interested in finding the block $\bar{b}$ that minimizes $\mathrm{SAD}_{\Delta\mathrm{IP}}$ in a proper range window of size R centered in b. We can compute only $S^V + S^H$ (saving computation as reported below) and compare this value with the best $\mathrm{SAD}_{\Delta\mathrm{IP}}$ previously computed ($\overline{\mathrm{SAD}_{\Delta\mathrm{IP}}}$). If $S^V + S^H$ is greater than $\overline{\mathrm{SAD}_{\Delta\mathrm{IP}}}$, further computations are not needed. From (12) we have

$$\overline{\mathrm{SAD}_{\Delta\mathrm{IP}}} \le S^V + S^H \le \mathrm{SAD}_{\Delta\mathrm{IP}}. \tag{13}$$

On the contrary (if $S^V + S^H$ is less than $\overline{\mathrm{SAD}_{\Delta\mathrm{IP}}}$), we have to compute the error function (see Algorithm 1). The overall complexity per block of the proposed approach becomes

$$(2R+1)^2(N-2) + \overline{F_N}(2(2N-3)) + 8N^2 \tag{14}$$

where $\overline{F_N}$ is the number of failures (i.e., when $\overline{\mathrm{SAD}_{\Delta\mathrm{IP}}} \ge S^V + S^H$). The computation of $S^V + S^H$ has a cost of $(2R+1)^2(N-2)+2N^2$. The sum of adjacent elements has been computed only one time saving a lot of computation. It is important to note that the overall computational saving strictly depends on the $\overline{F_N}$ value. Moreover, the order in which the error function is evaluated impacts the final $\overline{F_N}$ value. The smart selection of the initial motion vector can effectively improve the performance of the algorithm with respect to the simple scanning top-bottom, left-right (Fig. 3). Due to the motion continuity (neighboring blocks typically have similar motion vectors), we consider as initial motion estimate the motion vector obtained in the previous algorithm iteration (it typically corresponds to a neighboring block). In the following section, we report some numerical results obtained on real cases that justify the usage of the optimization strategy.

Algorithm 1 Optimized selection of the best matching block

```
Input: A block $b^t$ of the frame $I^t$ and the frame $I^{t+1}$
Output: The block $\overline{b^{t+1}}$ of frame $I^{t+1}$ with the best match with $b^t$
begin
    $\overline{\mathrm{SAD}_{\Delta\mathrm{IP}}} = Inf$
    for $i = -R$ to $R$ do
        for $j = -R$ to $R$ do
            compute $S^V(b^t, b_{ij}^{t+1})$
            compute $S^H(b^t, b_{ij}^{t+1})$
            if $\overline{\mathrm{SAD}_{\Delta\mathrm{IP}}} > S^V(b^t, b_{ij}^{t+1}) + S^H(b^t, b_{ij}^{t+1})$ then
                compute $\mathrm{SAD}_{\Delta\mathrm{IP}}(b^t, b_{ij}^{t+1})$
                if $\mathrm{SAD}_{\Delta\mathrm{IP}}(b^t, b_{ij}^{t+1}) < \overline{\mathrm{SAD}_{\Delta\mathrm{IP}}}$ then
                    $\overline{\mathrm{SAD}_{\Delta\mathrm{IP}}} = \mathrm{SAD}_{\Delta\mathrm{IP}}(b^t, b_{ij}^{t+1})$
                    $\overline{b^{t+1}} = b_{ij}^{t+1}$
end
```

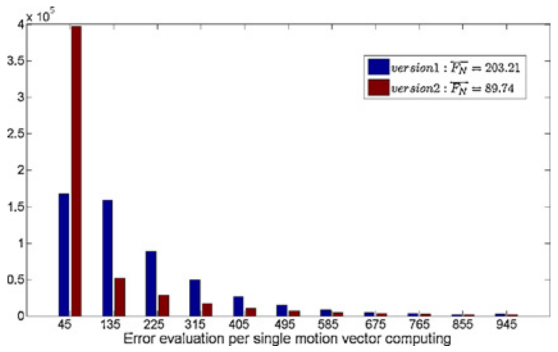

Fig. 3. Histograms of the number of error function evaluation relative to different strategies: simple scanning top-bottom, left-right (blue); initial motion estimate chosen as the motion vector obtained in the previous iteration (red).

B. Block-Based Estimator Comparisons

For sake of comparison, we have employed some well-known block-based local motion estimators: FSA, three-step search algorithm (TSS), and [31]. FSA is an exhaustive approach in which all the possible correspondences are tested. The error measure often adopted is the SAD defined as follows:

$$\mathrm{SAD} = \sum_{i=1}^{N}\sum_{j=1}^{N} |b_1(i,j) - b_2(i,j)| \tag{15}$$

where b_1 and b_2 are two $(N \times N)$ blocks to be compared. Let R the search range, there are $(2R+1)^2$ possible correspondences in which SAD is computed. The minimum SAD value hence indicates the best correspondence. FSA typically performs pretty well but it has a high computational cost. Per block we have to perform about $(2R+1)^2 2N^2$ additions.

On the contrary, TSS, in order to considerably reduce the computational cost, does not consider all the possible correspondences performing a coarse to fine searching strategy. It starts with a step size equal to half search range evaluating nine points (one in the center and eight in the search area boundaries). At the end of each step, the step size is halved and the minimum error point becomes the novel center. The algorithm ends when step size is equal to one. The computational complexity per block, considering SAD as matching function, is $9(log_2(R/2)+1)2N^2$, where R is the search range and N is the block size. TSS can be misled by local minima hence degrading their performances.

The authors in [31] proposed an exhaustive approach (all the possible correspondences are tested) with a 1-D matching function based on vertical integral projections. The computational complexity per block, computed avoiding redundant operations, is $(2R+1)^2(2N-1)+2N^2$. In our experiments, we have also considered an extended version that considers both horizontal and vertical projections (the complexity is twice with respect to the simplest version).

In Table I, the overall number of additions relative to the local motion estimators has been reported. As pointed above, the value of $\overline{F_N}$ is the key factor of the proposed searching strategy. To properly evaluate the performance, an evaluation dataset has been built. Real videos can be acquired, of course, with different cameras and environmental conditions (e.g., illumination). A proper experimental pipeline has been hence designed. In order to cope with scene variability, our tests have been performed on the Oliva and Torralba dataset [34] made up of 2688 images (256 × 256 pixels each) belonging to different classes: tall buildings, inside city, street, highway, coast, open country, mountain, and forest. Each image I has been modified through a similarity transformation: two shifts (T_x, T_y), one rotation angle (θ), and a zoom factor (λ). This transformation associates a point (x, y) in frame I with a point $(\widetilde{x}, \widetilde{y})$ in frame $\widetilde{I}$ as follows:

$$\begin{cases} \widetilde{x} = x\lambda\cos\theta - y\lambda\sin\theta + T_x \\ \widetilde{y} = x\lambda\sin\theta + y\lambda\cos\theta + T_y \end{cases} \tag{16}$$

TABLE I
COMPUTATIONAL COMPLEXITY COMPARISON

	No. of Additions
FSA	$(2R+1)^2 2N^2$
TSS	$9(log_2(R/2)+1)2N^2$
[31]	$(2R+1)^2(2N-1)+2N^2$
[31] with horizontal and vertical projections	$2((2R+1)^2(2N-1)+2N^2)$
Proposed approach	$(2R+1)^2 2(2N-3)+6N^2$
Proposed optimized approach (14)	$(2R+1)^2(N-2)+\overline{F_N}(2(2N-3))+8N^2$

The overall number of additions has been reported as a function of the search range (R), the block size (N), and the $\overline{F_N}$ parameter (14).

TABLE II
BLOCK-BASED APPROACHES COMPARISON IN TERMS OF ADDITION NUMBER

	No. of Additions
FSA	557 568
TSS	18 432
[31]	34 271
[31] with horizontal and vertical projections	68 542
Proposed approach	64 698
Optimized proposed approach (14)	22 499

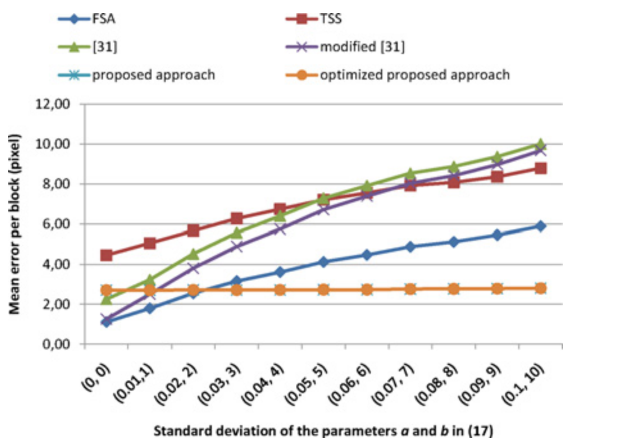

Fig. 4. Mean error per block $\overline{E}$ at increasing levels of illumination changes. The performance of the proposed approach does not depend on the illumination changes. On the contrary, the other approaches considerably degrade their performances in presence of this kind of disturb.

where λ, θ, T_x, and T_y are random variables drawn from Gaussian distributions with means 1, 0, 0, and 0, and standard deviations equal to 0.005, 0.5, 4, and 4, respectively. Afterward, a linear illumination change together with an additive Gaussian noise have been applied as follows:

$$\overline{I}(i,j) = a\widetilde{I}(i,j) + b + \eta(i,j) \tag{17}$$

where a and b are random variable drawn from zero mean Gaussian distributions and η is a zero mean Gaussian noise with standard deviation equal to 0.01. Note that a and b are constant over the image whereas η is variable (typically has a different value per pixel). We have then obtained 2688 pairs of corresponding images to be used for comparisons. In order to underline the robustness of the proposed approach with respect to illumination changes, several tests have been conducted considering various illumination change intensities (i.e., increasing standard deviations of a and b parameters). The mean error per block, used as evaluation criterion, has been computed from the cumulative error E defined as follows:

$$E = \sum_{k=1}^{N_f}\sum_{i=1}^{B_r}\sum_{j=1}^{B_c} err(i,j,k) \tag{18}$$

$$err(i,j,k) = |v_x(i,j,k) - s_x(i,j,k)| + |v_y(i,j,k) - s_y(i,j,k)| \tag{19}$$

where N_f is the number of frames used in the test (i.e., 2688), B_r and B_c are the block number per row and column in each frame, and (v_x, v_y), (s_x, s_y) are the measured and estimated local motion vector components, respectively. The mean error per block $\overline{E}$ is obtained just averaging with respect to the total number of blocks ($N_f B_r B_c$). As reported in Fig. 4 and Table II, the proposed approach obtains satisfactory results outperforming, in almost all cases, the other techniques both in terms of measured error $\overline{E}$ and number of additions.

In our tests, we have obtained a $\overline{F_N}$ value of 89.74. Considering the parameters values used in the tests ($N = R = 16$), from (14), 22 499 additions have to be done. The measured results confirm the effectiveness of the developed block-based local motion estimator having a computational complexity very close to TSS but performances usually better than FSA (Fig. 4) in presence of illumination changes. It is important to notice that the poor results of [31] are mainly due to the illumination changes that highlight one of the weaknesses of pure integral projections matching.

III. Voting Approach

Videos acquired by common imaging devices (PDAs, mobile phones, etc.) can dramatically vary in their content (indoor, outdoor, etc.) and involved environmental conditions (e.g., illumination). Making robust video stabilization algorithms able to work properly in almost all real conditions becomes hence a difficult task. Many factors can degrade, if not properly managed, the performances of video stabilization algorithms. For instance, in presence of homogeneous regions, periodic patterns, and fast illumination changes, local motion estimators, sometimes, produce wrong vectors. Moreover, the movement of the objects in the scene can mislead the global motion estimation because the corresponding vectors, even if correctly computed, do not describe camera movements. Finally, zooming and forward walking of the user can create some difficulties if they are not taken into account in the motion model (e.g., a translational motion model). Different solutions have been adopted in order to cope with these problems. We recall that at this stage, a set of local motion vectors describing correspondences between adjacent frames has been derived. Some approaches remove outliers (wrong vectors or vectors that do not describe the global motion) by using some heuristics (motion continuity, homogeneity of the regions, matching effectiveness, etc.) [5], [35], [36]. The inliers are then used as input of a least square estimator (sensitive to outliers). Other approaches simply make use of a robust estimator (e.g., Ransac [37], M-estimator [38], and others) [6], [17], [39].

Some approaches based on a voting strategy have been developed considering simple 2-D models: translational [8] and three parameter model (two shifts and a scale factor) [40]. Our approach generalizes these techniques considering a 2-D similarity model (two shifts, a scale factor, and a rotation). Starting from (16), by means of some simple mathematical manipulations, each pair of corresponding points (x, y) and $(\widetilde{x}, \widetilde{y})$ is related by the following (see Appendix A):

$$(\widetilde{x} - T_x)^2 + (\widetilde{y} - T_y)^2 = \lambda^2(x^2 + y^2). \tag{20}$$

Each pair (provided by the local motion estimator) represents a cone in the parameter space (T_x, T_y, λ). The unknown parameter values $(\overline{T_x}, \overline{T_y}, \overline{\lambda})$ can be then obtained considering the densest region of the space (Fig. 5) making use of a 3-D histogram. The motion vectors belonging to a $3 \times 3 \times 3$ neighborhood of the involved triplet are then used to compute the rotational parameter (θ) with a voting-based approach by means of 1-D histogram (see Appendix B) as follows:

$$\theta = \arctan \frac{x\widetilde{y} - \widetilde{x}y + y\overline{T_x} - x\overline{T_y}}{\widetilde{y}y + \widetilde{x}x - x\overline{T_x} - y\overline{T_y}}. \tag{21}$$

In order to obtain higher accuracy (limited by the binning process), we have performed a parabolic interpolation of the

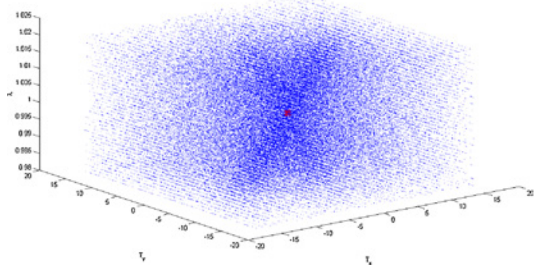

Fig. 5. Each pair of corresponding coordinates $(x, y), (\widetilde{x}, \widetilde{y})$ provided by the local motion estimator describes a curve in the parametric space (20). The curves related to correct pairs (i.e., describing the correct motion) should then intersect in a small region. The densest region in the parameter space represents the region voted by the highest number of pair of corresponding coordinates. The estimated triplet $(\overline{T_x}, \overline{T_y}, \overline{\lambda})$ has been represented with a red cross.

TABLE III

VOTING APPROACHES COMPARISON

	Original	Voting (a)	Voting (b)	Voting (c)	Voting (d)
Indoor1	35.5	40.3	40.9	40.8	42.0
Indoor2	39.3	41.3	41.7	42.3	43.2
Outdoor1	31.8	34.0	34.1	34.5	34.4
Outdoor2	25.1	33.8	33.4	33.5	33.9
Illumination changes	40.7	43.3	43.4	43.4	43.7
Zoom	26.8	29.0	28.9	29.0	29.0
Average	**33.2**	**37.0**	**37.1**	**37.2**	**37.7**

(a) Pixel accuracy, (b) final LS estimation and pixel accuracy, (c) subpixel accuracy, and (d) subpixel accuracy and final LS estimation. Notice that pixel and subpixel accuracy refers to the motion vectors provided by the local motion estimator. The global ITF average is represented in bold.

histogram points adjacent (one per side) to the most voted value, taking the coordinate of the parabola vertex estimated as $\overline{\theta}$. Similar interpolations, even if in higher dimension, have been performed in the 3-D histogram.

At the end of these voting steps, the estimation of the unknown parameter $(\overline{\lambda}, \overline{\theta}, \overline{T_x}, \overline{T_y})$ is obtained. However, better accuracy can be achieved (Section IV) considering these values only to filter out outliers. The filtering is performed in the 3-D (T_x, T_y, λ) and 1-D histogram (θ) retaining only the motion vectors voting for the bins adjacent to the best one. Finally, inliers are used as input to the final least square estimation. It is important to note that [8] uses a 2-D histogram (two parameters to be estimated) whereas [40] uses a 3-D histogram (three parameters to be estimated). Our approach, properly combining (16), makes use of a 3-D histogram (T_x, T_y, λ) followed by a 1-D histogram (θ) instead of a 4-D histogram to estimate four parameters obtaining hence a considerable computational and memory resource saving.

IV. Experimental Results

In this section, we show the effectiveness of our approach through a series of experiments conducted on videos captured by handheld digital cameras with different contents (indoor, outdoor) and conditions (illumination changes, moving objects, variable zooming factor, periodic patterns). These videos have a resolution of 640×480 (*Indoor1*, *Indoor2*, *Outdoor1*, *Outdoor2*, *Illumination changes*, *Zoom*, *MovingObject_1*, *MovingObject_2*, *Periodic_1D*, *Periodic_2D*) and 720×480 (*VHall*, *VBranches*) pixels. Numerical evaluation of the quality of the video stabilization is fulfilled using interframe transformation fidelity (ITF) [2] as error measure

$$\mathrm{ITF} = \frac{1}{N_{\mathrm{frame}} - 1} \sum_{k=1}^{N_{\mathrm{frame}}-1} \mathrm{PSNR}(k) \tag{22}$$

where PSNR(k) is the peak signal-to-noise ratio between two consecutive frames $(k, k+1)$ and N_{frame} is the video frames number. PSNR measures how much an image is similar to another one, hence it is useful to quantitatively evaluate a stabilized sequence by the algorithm by simply measuring the similarity of consecutive frames in the final sequence. Typically, the stabilized sequence has a higher ITF value than the original one even if it must be carefully used. This measure can be misled by many factors: big moving objects in the scene, scene changes, periodic patterns, and others. The evaluation of the videos containing moving objects and periodic patterns has been hence visually performed (see Figs. 6–9 and http://iplab.dmi.unict.it/download/Video/TCSVT2011).

Several versions of our voting approach have been compared (Table III). They differ with respect to the presence of the final least square (LS) estimation step and the subpixel accuracy of the block-based local motion estimator (Section II). Subpixel accuracy has been obtained by using a simple parabolic interpolation around the best match. This step, performed only on the already computed motion vectors (not on the image pixels), does not degrade the computational performances of the local estimator. It should be noted that in our tests block size $N = 16$ and the search range $R = 16$. All these approaches make use of a 3-D histogram (T_x, T_y, λ) of $33 \times 33 \times 21$ bins and a 1-D histogram (θ) of 61 bins. T_x and T_y range from -16 to 16 (it depends on the local estimator), λ from 0.98 to 1.02°, and θ from -3 to 3°. As previously stated, the final LS estimation step provides higher accuracy without sacrificing robustness in presence of moving objects and periodic patterns. Moreover, as expected, subpixel information increases the accuracy of the method.

The proposed approach has been compared with some standard techniques ([9], [10], Ransac [37]), with two state-of-the-art algorithms [5], [8] and a freeware software (Deshaker[1]). Ransac [5], [8] take as input the vectors provided by our local motion estimator (subpixel accuracy). Moreover, to cope with randomness, each Ransac entry has been computed as the average of five experiments. Both [9] and [10] have been reimplemented considering four subwindows of 128×128 pixels each.

Our approach outperforms the other techniques, obtaining (on average) higher accuracy (Table IV). The robustness has been tested considering four critical sequences containing moving objects and periodic patterns.

1) *MovingObject_1*: Some fruits rolling over a table (Fig. 6).
2) *MovingObject_2*: A big textured object slowly entering in the scene (Fig. 7).

[1]Available at http://compression.ru/video/deshaker/index_en.html.

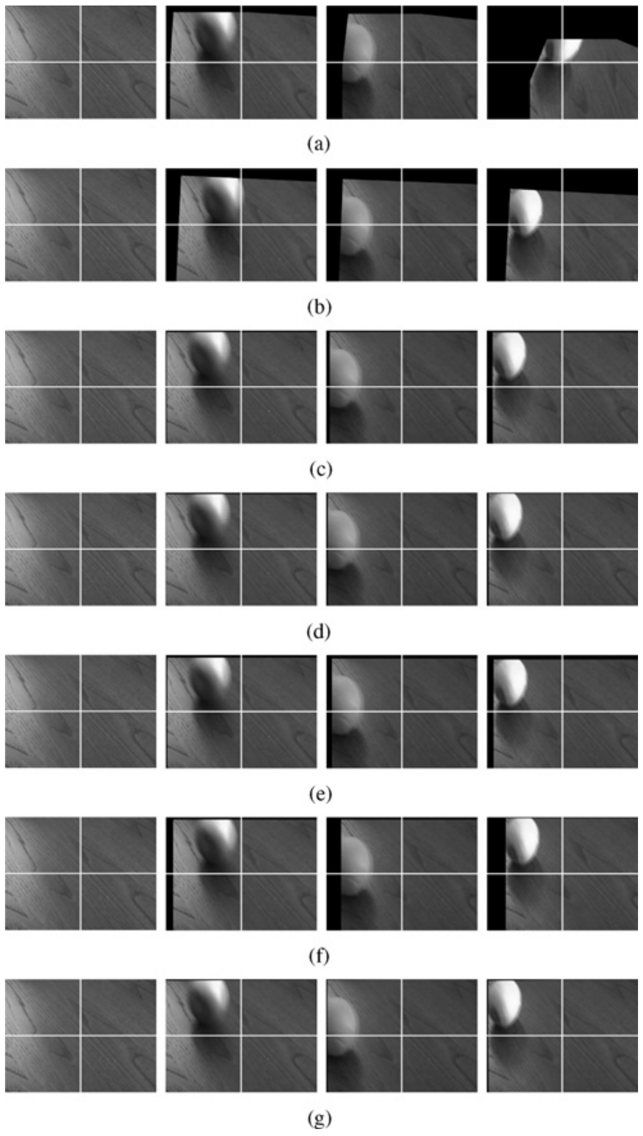

Fig. 6. Video containing some fruits rolling over a table. Battiato *et al.* [5], Deshaker, and [10] do not work properly (vectors belonging to moving objects are not filtered out). Although Ransac and [9] perform better than Deshaker and Battiato *et al.* [5], cumulative error is present (see initial and final frame). Our approach (voting) and [8] are not deceived by the passing of the fruits. A white cross has been added for better visualization. (a) Battiato *et al.* [5]. (b) Deshaker. (c) Ransac. (d) Chen *et al.* [8]. (e) Ko *et al.* [9]. (f) Ko *et al.* [10]. (g) Proposed approach.

3) *Periodic_1D*: A video containing a 1-D periodic pattern (Fig. 8).
4) *Periodic_2D*: A video containing a 2-D periodic pattern (Fig. 9).

Battiato *et al.* [5], due to its filtering rules, is not able to work properly in presence of textured moving objects on a homogeneous background. Moreover, its limits in presence of periodic patterns have been already shown in [41]. Deshaker fails stabilizing all the challenging sequences: vectors belonging to moving objects are not filtered out and it is not able to cope with periodic patterns. Ransac performs better than Deshaker, but cumulative error is present in the first sequence (see initial and final frame) and it does not work properly in the second and third one. Although we set the maximum trials number (Ransac parameter) to 1000, in these critical conditions, this value sometimes is reached without convergence. Hence, its output (the best found) can be wrong. Chen *et al.* [8]–[10] have shown a certain degree of robustness but their performances (i.e., accuracy) are limited by the adopted translational model (e.g., rotations are not taken into account). Moreover, both [9] and [10] suffer in presence of sudden illumination changes (*Illumination changes*) and homogeneous backgrounds (*Indoor2*). Our approach is not deceived by these moving objects (just a bit at the end of the second sequence) and it is able to work properly even in presence of periodic patterns (see Table V). The voting filtering step (Section III) provides high robustness whereas the final LS step high accuracy. Notice that our approach is totally deterministic, no random criteria are used.

TABLE IV
ITF COMPARISON

	[5]	Ransac	Deshaker	[8]	Voting	[9]	[10]
Indoor1	39.6	41.1	40.2	40.9	42.0	38.2	38.2
Indoor2	42.8	41.1	41.6	40.3	43.2	38.8	39.7
Outdoor1	34.1	34.3	34.0	33.9	34.4	33.6	33.4
Outdoor2	33.8	34.0	33.5	33.3	33.9	32.3	33.2
Illumination changes	42.8	43.8	43.1	43.2	43.7	42.0	41.9
Zoom	28.7	29.0	28.4	28.6	29.0	28.0	28.0
Average	**37.0**	**37.2**	**36.8**	**36.7**	**37.7**	**35.5**	**35.7**

Our approach obtains (on average) the best results (i.e., accuracy). The global ITF average is represented in bold.

TABLE V
ROBUSTNESS COMPARISON

Video	[5]	Ransac	Deshaker	[8]	Voting	[9]	[10]
1	F	**OK**	F	**OK**	**OK**	**OK**	F
2	**OK**	F	F	F	**OK**	**OK**	**OK**
3	F	F	F	F	**OK**	**OK**	**OK**
4	F	**OK**	F	**OK**	**OK**	**OK**	**OK**

Our approach works properly even in challenging conditions: big moving objects and periodic patterns. In the table, F stands for failure.

Further tests have been performed considering two sequences with high jitter and complex 3-D camera movements.[2] To deal with intentional movements, a simple motion vector integrator has been applied to the involved methods. It should be noted that the design of a novel technique of motion filtering (i.e., discrimination of intentional motion from the unwanted motion) is out of the scope of this paper. As can be easily seen from Table VI, our approach works properly even in the aforementioned conditions. Deshaker has not been included in these tests. This software has its own motion filtering method (it was disabled in the tests of Tables IV and V), hence a fair comparison cannot be done.

V. COMPUTATIONAL COMPLEXITY

As already stated in Section II, the complexity of the local estimator depends on the parameter F_N of (14). As reported

[2]Available at http://rvsn.csail.mit.edu/pv/data/input-video.

TABLE VI
ITF COMPARISON ON SEQUENCES CONTAINING HIGH JITTER AND COMPLEX 3-D MOVEMENTS

Video	[5]	Ransac	[8]	Voting	[9]	[10]
VHall	28.8	28.8	28.4	28.7	28.3	28.3
VBranches	22.2	22.2	21.5	22.2	21.4	21.5
Average	**25.5**	**25.5**	**25.0**	**25.5**	**24.9**	**24.9**

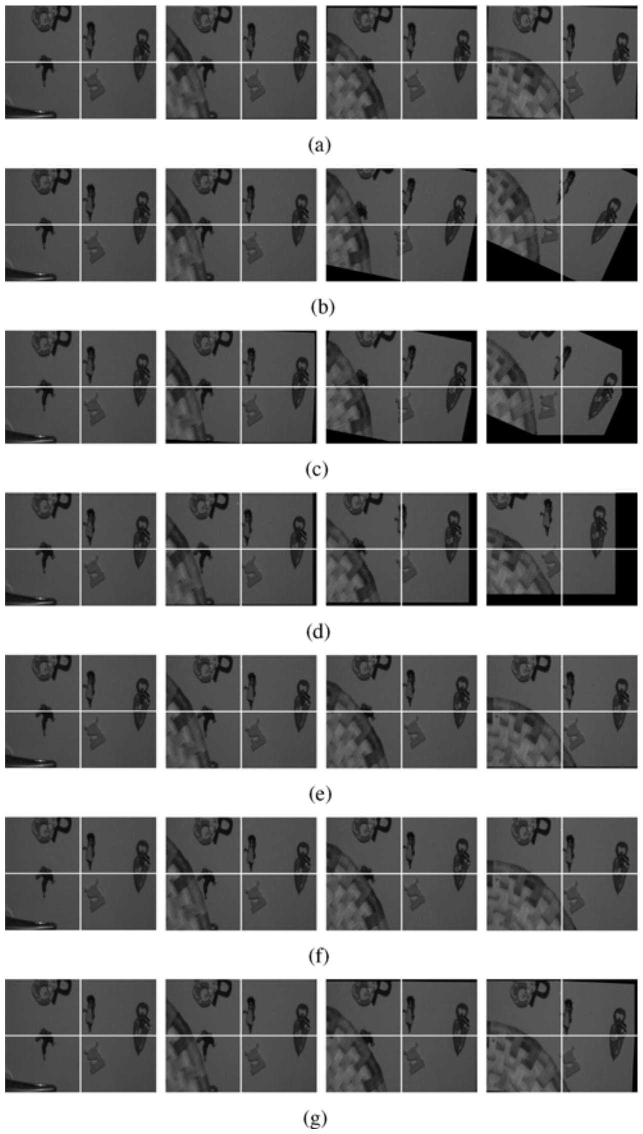
Fig. 7. Video containing a big textured object slowly entering in the scene. Deshaker, Ransac, and Chen *et al.* [8], in this critical condition, are not able to deal with outliers. On the contrary, [9], [10], Battiato *et al.* [5], and the proposed approach properly stabilize almost all video (they are mislead, just a bit, at the end of the sequence). A white cross has been added for better visualization. (a) Battiato *et al.* [5]. (b) Deshaker. (c) Ransac. (d) Chen *et al.* [8]. (e) Ko *et al.* [9]. (f) Ko *et al.* [10]. (g) Proposed approach.

in Table VII, the optimization procedure obtains satisfactory results in almost all the sequences used in our tests. Although the proposed optimization strategy does not work well in presence of homogeneous regions (sequence *Indoor2*), even in the worst case, the final addition number is lower than that obtained without performing any optimization.

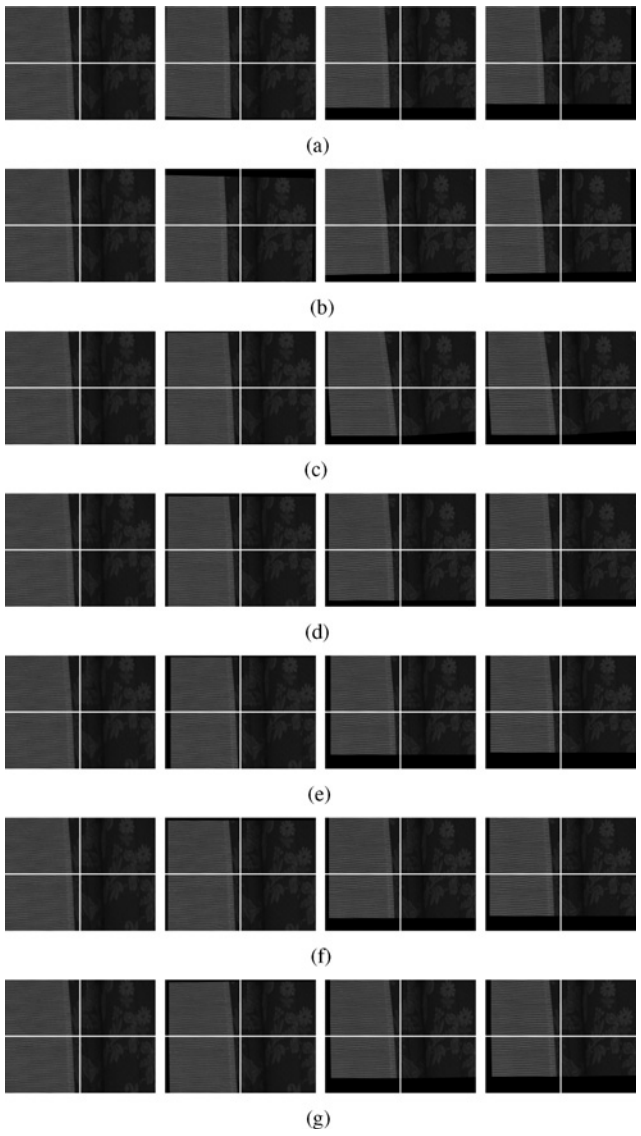
Fig. 8. Video containing a 1-D periodic pattern. Battiato *et al.* [5], Deshaker, Ransac, and Chen *et al.* [8], in this critical condition, are not able to cope with outliers. On the contrary, our approach, [9] and [10], work properly. A white cross has been added for better visualization. (a) Battiato *et al.* [5]. (b) Deshaker. (c) Ransac. (d) Chen *et al.* [8]. (e) Ko *et al.* [9]. (f) Ko *et al.* [10]. (g) Proposed approach.

TABLE VII
NUMBER OF ADDITIONS OF THE PROPOSED LOCAL ESTIMATOR WITH AND WITHOUT OPTIMIZATION (14)

Sequences	F_N	No. of additions with optimization	No. of additions without optimization
Indoor1	127	24 660	64 698
Indoor2	729	59 576	64 698
Outdoor1	44	19 846	64 698
Outdoor2	40	19 614	64 698
Illumination changes	64	21 006	64 698
Zoom	35	19 324	64 698
VHall	224	30 286	64 698
VBranches	44	19 846	64 698
Average	**163**	**26 770**	**64 698**

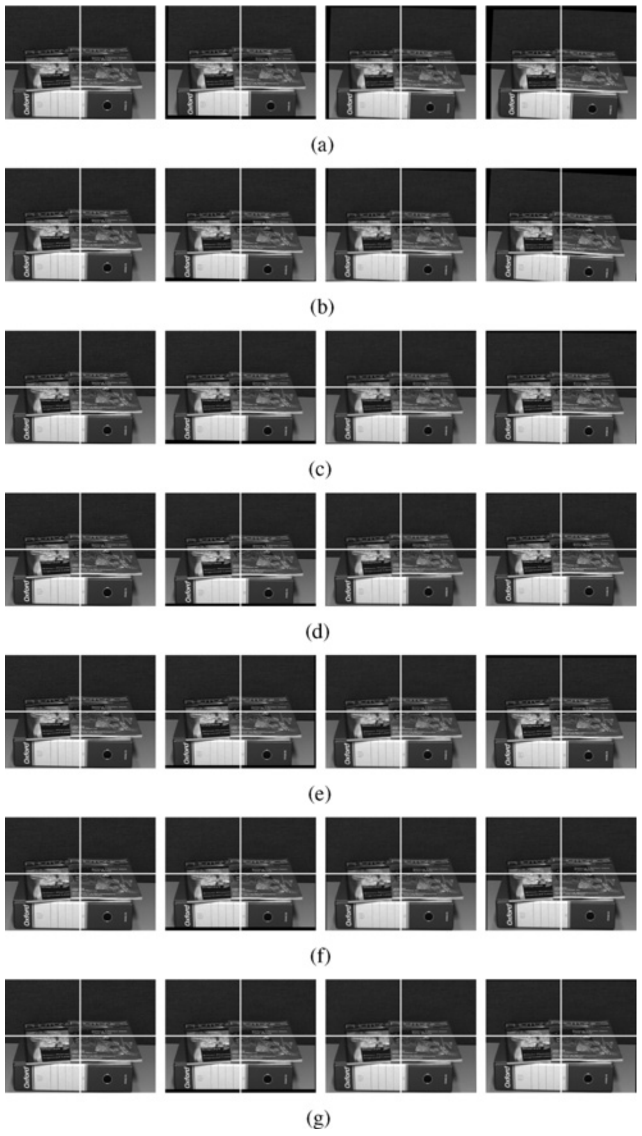

Fig. 9. Video containing a 2-D periodic pattern. This pattern, generating wrong matchings, deceives both Battiato *et al.* [5] and Deshaker. On the contrary, the other approaches work properly. A white cross has been added for better visualization. (a) Battiato *et al.* [5]. (b) Deshaker. (c) Ransac. (d) Chen *et al.* [8]. (e) Ko *et al.* [9]. (f) Ko *et al.* [10]. (g) Proposed approach.

In the following, both theoretical analysis and numerical results about the computational complexity of the global estimator are provided. The complexity of the proposed voting approach is proportional to the range of the translations to be estimated R_T (33 in our tests), to the considered scales N_S (21 in our tests), and to the number of involved motion vectors N_{mv}. To sum up, the complexity is $O(R_T N_S N_{mv})$. Considering a video graphics array (VGA) video (1200 blocks of 16×16 pixels), (20) has to be taken into account $33 \times 21 \times 1200$ times. In order to have a quantitative measure of the complexity of the algorithm, some tests have been performed on a Intel(R) Core(TM)2 Duo CPU 2.53 GHz. The proposed global motion estimator approach has been compared with other algorithms in terms of execution time. All the considered techniques have been implemented in MATLAB (version 7.11.0). Chen *et al.* [8] due to its simplicity (it considers only a translational model) is very fast (about 0.015 s per frame). The proposed voting approach takes about 0.3 s per frame whereas Battiato *et al.* [5] takes about 1 s per frame. Finally, Ransac takes on average 1.25 s per frame. It should be noted that Ransac has a high variability in terms of execution times ranging from 0.05 to 4.5 s per frames in the challenging videos. On the contrary, the other approaches show a low variability. To sum up, the proposed approach is pretty fast, the best considering the techniques with higher accuracy (more complex adopted model), and with an almost constant execution time and memory usage per frame (very useful properties for real-time system on embedded devices).

VI. Conclusion

In this paper, we have proposed a fast and robust image alignment algorithm for video stabilization purposes. Our contribution has been twofold. A block-based local motion estimator designed to be as fast as possible maintaining at the same time a high accuracy. This result has been obtained combining together the exhaustive search strategy with an integral projection-based error function. Motion vectors provided by the local estimator have been then used to compute the global interframe motion parameter through a voting-based approach. The effectiveness of our approach has been demonstrated through a series of experiment in critical conditions and comparisons with standard and recent techniques. Future work will be devoted to perform a temporal analysis able to include the information coming from previous iterations in the filtering step. Moreover, to further validate the performance of the proposed method in terms of execution time, we plan to do perform some tests on a ARM simulator.

Appendix A
Mathematical Derivation (20)

In our approach, we have considered a 2-D similarity transformation: two shifts (T_x, T_y), one rotation angle (θ), and a zoom factor (λ). This transformation associates a point (x, y) in frame I with a point $(\widetilde{x}, \widetilde{y})$ in frame $\widetilde{I}$ as follows:

$$\widetilde{x} = x\lambda\cos\theta - y\lambda\sin\theta + T_x \tag{23}$$
$$\widetilde{y} = x\lambda\sin\theta + y\lambda\cos\theta + T_y. \tag{24}$$

Moving T_x and T_y on the left side and squaring

$$(\widetilde{x} - T_x)^2 = \lambda^2(x\cos\theta - y\sin\theta)^2 \tag{25}$$
$$(\widetilde{y} - T_y)^2 = \lambda^2(x\sin\theta + y\cos\theta)^2. \tag{26}$$

Combining together (sum) (25) and (26)

$$\begin{aligned}(\widetilde{x} - T_x)^2 + (\widetilde{y} - T_y)^2 &= \lambda^2[(x\cos\theta - y\sin\theta)^2 \\ &\quad +(x\sin\theta + y\cos\theta)^2] \\ &= \lambda^2[x^2\cos^2\theta + y^2\sin^2\theta - 2xy\cos\theta\sin\theta + \\ &\quad +x^2\sin^2\theta + y^2\cos^2\theta + 2xy\sin\theta\cos\theta] \\ &= \lambda^2[(x^2 + y^2)(\cos^2\theta + \sin^2\theta)].\end{aligned} \tag{27}$$

Making use of the trigonometric equality $\cos^2\theta + \sin^2\theta = 1$, we finally obtain (20)

$$(\widetilde{x} - T_x)^2 + (\widetilde{y} - T_y)^2 = \lambda^2(x^2 + y^2). \tag{28}$$

APPENDIX B MATHEMATICAL DERIVATION (21)

Let $\widetilde{a} = \lambda\cos\theta$ and $\widetilde{b} = \lambda\sin\theta$. Equations (23) and (24) become

$$\widetilde{x} = x\widetilde{a} - y\widetilde{b} + T_x \tag{29}$$
$$\widetilde{y} = x\widetilde{b} + y\widetilde{a} + T_y. \tag{30}$$

Solving (29) and (30) with respect to $\widetilde{a}$ and $\widetilde{b}$

$$\widetilde{a} = \frac{\widetilde{y}y + \widetilde{x}x - xT_x - yT_y}{x^2 + y^2} \tag{31}$$
$$\widetilde{b} = \frac{x\widetilde{y} - \widetilde{x}y + yT_x - xT_y}{x^2 + y^2}. \tag{32}$$

We are interested to find the ratio $\widetilde{b}/\widetilde{a}$ (from definition equals to $\tan\theta$)

$$\frac{\widetilde{b}}{\widetilde{a}} = \frac{x\widetilde{y} - \widetilde{x}y + yT_x - xT_y}{\widetilde{y}y + \widetilde{x}x - xT_x - yT_y}. \tag{33}$$

We finally obtain (21)

$$\theta = \arctan\frac{x\widetilde{y} - \widetilde{x}y + yT_x - xT_y}{\widetilde{y}y + \widetilde{x}x - xT_x - yT_y}. \tag{34}$$

REFERENCES

[1] P.-M. Jodoin, J. Konrad, V. Saligrama, and V. Veilleux-Gaboury, “Motion detection with an unstable camera,” in *Proc. ICIP*, Oct. 2008, pp. 229–232.

[2] L. Mercenaro, G. Vernazza, and C. Regazzoni, “Image stabilization algorithms for video-surveillance application,” in *Proc. ICIP*, 2001, pp. 349–352.

[3] K.-Y. Lee, Y.-Y. Chuang, B.-Y. Chen, and M. Ouhyoung, “Video stabilization using robust feature trajectories,” in *Proc. IEEE Int. Conf. Comput. Vision*, Sep.–Oct. 2009, pp. 1397–1404.

[4] S. Battiato and R. Lukac, “Video stabilization techniques,” in *Encyclopedia of Multimedia*. New York: Springer-Verlag, Oct. 2008, pp. 941–945.

[5] S. Battiato, A. R. Bruna, and G. Puglisi, “A robust block based image/video registration approach for mobile imaging devices,” *IEEE Trans. Multimedia*, vol. 12, no. 7, pp. 622–635, Nov. 2010.

[6] S. W. Jang, M. Pomplun, G. Kim, and H. I. Choi, “Adaptive robust estimation of affine parameters from block motion vectors,” *Image Vision Comput.*, vol. 23, no. 14, pp. 1250–1263, Dec. 2005.

[7] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H.-Y. Shum, “Full-frame video stabilization with motion inpainting,” *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 28, no. 7, pp. 1150–1163, Jul. 2006.

[8] H. Chen, C.-K. Liang, Y.-C. Peng, and H.-A. Chang, “Integration of digital stabilizer with video codec for digital video cameras,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 7, pp. 801–813, Jul. 2007.

[9] S.-J. Ko, S.-H. Lee, and K.-H. Lee, “Digital image stabilizing algorithms based on bit-plane matching,” *IEEE Trans. Consum. Electron.*, vol. 44, no. 3, pp. 617–622, Aug. 1998.

[10] S.-J. Ko, S.-H. Lee, S.-W. Jeon, and E.-S. Kang, “Fast digital image stabilizer based on gray-coded bit-plane matching,” *IEEE Trans. Consum. Electron.*, vol. 45, no. 3, pp. 598–603, Aug. 1999.

[11] A. A. Yeni and S. Erturk, “Fast digital image stabilization using one bit transform based sub-image motion estimation,” *IEEE Trans. Consum. Electron.*, vol. 51, no. 3, pp. 917–921, Aug. 2005.

[12] O. Urhan and S. Erturk, “Single sub-image matching based low complexity motion estimation for digital image stabilization using constrained one-bit transform,” *IEEE Trans. Consum. Electron.*, vol. 52, no. 4, pp. 1275–1279, Nov. 2006.

[13] N.-J. Kim, H.-J. Lee, and J.-B. Lee, “Probabilistic global motion estimation based on Laplacian two-bit plane matching for fast digital image stabilization,” *EURASIP J. Adv. Signal Process.*, vol. 2008, no. 180582, pp. 1–10, 2008.

[14] M. Hansen, P. Anandan, K. Dana, G. van der Wal, and P. Burt, “Real-time scene stabilization and mosaic construction,” in *Proc. IEEE WACV*, Dec. 1994, pp. 54—62.

[15] Z. Zhu, G. Xu, Y. Yang, and J. S. Jin, “Camera stabilization based on 2.5D motion estimation and inertial motion filtering,” in *Proc. IEEE Int. Conf. Intell. Vehicles*, vol. 2. Oct. 1998, pp. 329–334.

[16] A. Adams, N. Gelfand, and K. Pulli, “Viewfinder alignment,” *Comput. Graphics Forum*, vol. 2, no. 27, pp. 597–606, 2008.

[17] C. R. del Blanco, F. Jaureguizar, L. Salgado, and N. García, “Automatic feature-based stabilization of video with intentional motion through a particle filter,” in *Proc. ACIVS*, LNCS 5259. 2008, pp. 356–367.

[18] J. Yang, D. Schonfeld, and M. Mohamed, “Robust video stabilization based on particle filter tracking of projected camera motion,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 7, pp. 945–954, Jul. 2009.

[19] A. Bosco, A. Bruna, S. Battiato, G. Bella, and G. Puglisi, “Digital video stabilization through curve warping techniques,” *IEEE Trans. Consum. Electron.*, vol. 54, no. 2, pp. 220–224, May 2008.

[20] S. Battiato, G. Gallo, G. Puglisi, and S. Scellato, “SIFT features tracking for video stabilization,” in *Proc. ICIAP*, 2007, pp. 825–830.

[21] S. Alliney and C. Morandi, “Digital image registration using projections,” *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 8, no. 2, pp. 222–233, Mar. 1986.

[22] X. Q. Gao, C. J. Duanmu, and C. R. Zou, “A multilevel successive elimination algorithm for block matching motion estimation,” *IEEE Trans. Image Process.*, vol. 9, no. 3, pp. 501–504, Mar. 2000.

[23] W. Li and E. Salari, “Successive elimination algorithm for motion estimation,” *IEEE Trans. Image Process.*, vol. 4, no. 1, pp. 105–107, Jan. 1995.

[24] C. Zhu, W. S. Qi, and W. Ser, “Predictive fine granularity successive elimination for fast optimal blockmatching motion estimation,” *IEEE Trans. Image Process.*, vol. 14, no. 2, pp. 213–221, Feb. 2005.

[25] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, “Motion compensated interframe coding for video conferencing,” in *Proc. Nat. Telecommun. Conf.*, vol. 4. 1981, pp. G5.3.1–G5.3.5.

[26] C. Zhu, X. Lin, and L. P. Chau, “Hexagon-based search pattern for fast block motion estimation,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 5, pp. 349–355, May 2002.

[27] S. Zhu and K.-K. Ma, “A new diamond search algorithm for fast block-matching motion estimation,” *IEEE Trans. Image Process.*, vol. 9, no. 2, pp. 287–290, Feb. 2000.

[28] Y. Koo and W. Kim, “An image resolution enhancing technique using adaptive sub-pixel interpolation for digital still camera system,” *IEEE Trans. Consum. Electron.*, vol. 45, no. 1, pp. 118–123, Feb. 1999.

[29] A. J. Crawford, H. Denman, F. Kelly, F. Pitié, and A. C. Kokaram, “Gradient based dominant motion estimation with integral projections for real time video stabilisation,” in *Proc. IEEE ICIP*, Oct. 2004, pp. 3371–3374.

[30] J. S. Kim and R. H. Park, “A fast feature-based block matching algorithm using integral projections,” *IEEE J. Select. Areas Commun.*, vol. 10, no. 5, pp. 968–971, Jun. 1992.

[31] C. Tu, T. Tran, J. Prince, and P. Topiwala, “Projection-based block matching motion estimation,” *Proc. SPIE Applicat. Digital Image Process. XXIII*, vol. 4115, pp. 374–384, Aug. 2000.

[32] S. Tai and Y. Lin, “Fast full-search block-matching algorithm for motion-compensated video compression,” in *Proc. ICPR*, 1996, pp. 914–918.

[33] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Patt. Recog.*, Apr. 2001, pp. 511–518.

[34] A. Oliva and A. Torralba, “Modeling the shape of the scene: A holistic representation of the spatial envelope,” *Int. J. Comput. Vision*, vol. 42, no. 3, pp. 145–175, 2001.

[35] W. H. Cho and K.-S. Hong, “Affine motion based CMOS distortion analysis and CMOS digital image stabilization,” *IEEE Trans. Consum. Electron.*, vol. 53, no. 3, pp. 833–841, Aug. 2007.

[36] J. Lee, Y. Park, S. Lee, and J. Paik, “Statistical region selection for robust image stabilization using feature-histogram,” in *Proc. ICIP*, Nov. 2009, pp. 1553–1556.

[37] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[38] P. J. Hube, *Robust Statistical Procedures*. New York: Society for Industrial and Applied Mathematics, 1996.

[39] Y. Shen, P. Guturu, T. Damarla, B. Buckles, and K. Namuduri, "Video stabilization using principal component analysis and scale invariant feature transform in particle filter framework," *IEEE Trans. Consum. Electron.*, vol. 55, no. 3, pp. 1714–1721, Aug. 2009.

[40] M. Soto, S. Maludrottu, and C. S. Regazzoni, "Fast and correspondence-less camera motion estimation based on voting mechanism and morton codes," in *Proc. IEEE ICIP*, Sep. 2010, pp. 745–748.

[41] S. Battiato, G. Puglisi, and A. R. Bruna, "Regular texture removal for video stabilization," in *Proc. ICPR*, Dec. 2008, pp. 1–4.

Giovanni Puglisi (M'11) was born in Acireale, Italy, in 1980. He received his degree in computer science engineering (summa cum laude) from Catania University, Catania, Italy, in 2005 and the Ph.D. degree in computer science in 2009.

He is currently a Contract Researcher with the Department of Mathematics and Computer Science, University of Catania. His current research interests include video stabilization and raster-to-vector conversion techniques. He is the author of several papers on these activities.

Sebastiano Battiato (M'04–SM'06) was born in Catania, Italy, in 1972. He received his degree in computer science (summa cum laude) from the University of Catania, Catania, in 1995, and the Ph.D. degree in computer science and applied mathematics from the University of Naples, Naples, Italy, in 1999.

From 1999 to 2003, he was the Leader of the Imaging Team at STMicroelectronics, Catania. He joined the Department of Mathematics and Computer Science, University of Catania, as an Assistant Professor in 2004, where he is currently an Associate Professor. His current research interests include image enhancement and processing, image coding, camera imaging technology, and multimedia forensics. He has published more than 90 papers in international journals, conference proceedings, and book chapters. He is a co-inventor on about 15 international patents, a reviewer for several international journals, and a regular member of numerous international conference committees.

Prof. Battiato has participated in many international and national research projects and was the Chair of several international events. He is an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEM FOR VIDEO TECHNOLOGY and the *SPIE Journal of Electronic Imaging* (Special Issue on "Digital photography and image compression"). He is also a Guest Editor of the following special issues: "Emerging methods for color image and video quality enhancement" published in the *EURASIP Journal on Image and Video Processing* in 2010, and "Multimedia in forensics, security, and intelligence," to be published in the IEEE MULTIMEDIA MAGAZINE. He is the Director (and Co-Founder) of the International Computer Vision Summer School, Sicily, Italy.