# CMOS digital image stabilization

**Article** *in* IEEE Transactions on Consumer Electronics · September 2007

DOI: 10.1109/TCE.2007.4341576 · Source: IEEE Xplore

# CMOS Digital Image Stabilization

Won-ho Cho, Dae-Woong Kim, and Ki-Sang Hong

**Abstract —** *In this paper, we present a CMOS digital image stabilization algorithm based on the characteristics of a rolling shutter camera. Due to the rolling shuttering mechanism of a CMOS sensor, a CMOS video frame shows CMOS distortions which are not observed in a CCD video frame, and previous video stabilization techniques cannot handle these distortions properly even though they can make a visually stable CMOS video sequence. In our proposed algorithm, we first suggest a CMOS distortion model. This model is based on a rolling shutter mechanism which provides a solution to solve the CMOS distortion problem. Next, we estimate the global image motion and the CMOS distortion transformation directly from the homography between CMOS frames. Using the two transformations, we remove CMOS distortions as well as jittering motions in a CMOS video. In the experimental results, we demonstrate that our proposed algorithm can handle the CMOS distortion problem more effectively as well as the jittering problem in a CMOS video compared to previous CCD-based digital image stabilization techniques*[1].

**Index Terms — CMOS Sensor, CMOS Distortion Model, Rolling Shutter, Digital Image Stabilization.**

## I. Introduction

The image sensor market has continued its strong growth, spurred by a booming camera phone and compact digital camera market. As a leading technology, CCDs have dominated the imaging sensor market but owing to the improved semiconductor technology, CMOS sensors are now dominating high-end digital cameras such as a D-SLR class with its advantage of high-integrity and low-manufacturing cost, unseating CCDs as a leading image sensor in the imaging market. Many camera companies are investing on the improved CMOS image sensor technology to use it in high-quality imaging products. But CMOS sensors have a different shuttering mechanism from CCD sensors, which is called a *rolling shutter*. Due to this mechanism, digital camcorders using a CMOS sensor show visually degraded video quality compared to CCD camcorders and its ugly effects can be easily observed in low-end camera models like PC cameras. Since these CMOS artifacts are usually combined with the user's clumsy hand motion, it is important to estimate the jittering motions in a video sequence to analyze these artifacts but they are not observed in a CCD image. Thus, previous CCD based Digital Image Stabilization (*DIS*) researches have focused on the removal of jittering motions in its video sequence [6]-[13].

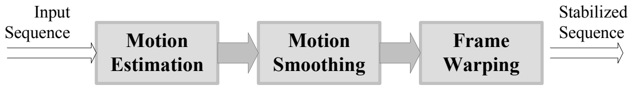


**Fig. 1. The block diagram of a general DIS technique**

Fig. 1 shows the block diagram of a typical DIS algorithm which consists of three parts: *global motion estimation*, *motion smoothing* and *frame warping*. The DIS technique, which has been studied for many years, makes up for user's clumsiness and stabilized the video clip's visual. Previous DIS researches have studied the method of removing jittering motions in video frames taken by a CCD camcorder, which now appear as plugin-type shareware [4] of some video editing programs for user's convenience. The motion estimation in Fig. 1 is to estimate a global image motion between frames which is a crucial part in DIS algorithm. The global image motion, for example, estimated by block matching [6]-[9] and/or optical flow methods [10]-[11], can be usually modeled by parametric 2D motion models. Also, the image registration techniques were used, which estimate warping parameters of adjacent frames and smooth them by Kalman filtering [13] or other smoothing methods [12]. Those previous approaches have implicitly assumed a CCD image sensor.

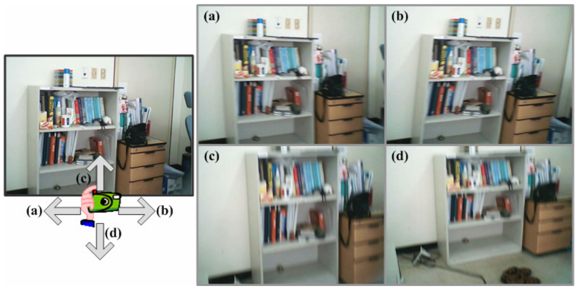


**Fig. 2. The motion distortions of CMOS web camera images: the camera is moving in the (a) left, (b) right, (c) up and (d) down directions.**

Unlike a CCD sensor, a CMOS sensor can cause typical distortions in its captured image. For example as in Fig. 2, vertical lines are slanted when conducting a fast horizontal motion, and scene objects in its image are vertically squeezed and stretched when moving up and down. This is due to the rolling shutter mechanism of a CMOS sensor. The rolling shutter captures the lines of sensor arrays one after another at slightly different times, which can cause ugly effects in a CMOS image formation.

[1] Won-ho Cho is with Division of Electrical and Computer Engineering, POSTECH, Pohang, Gyungbuk, S. Korea (e-mail: ellescho@postech.ac.kr).

Dae-woong Kim is with Samsung electronics Corp., Yongin, Gyunggi, S. Korea (e-mail: dwbear.kim@samsung.com).

Ki-Sang Hong is with Division of Electrical and Computer Engineering, POSTECH, Pohang, Gyungbuk, S. Korea (e-mail: hongks@postech.ac.kr).

Contributed Paper
Original manuscript received May 23, 2007
Revised manuscript received June 26, 2007

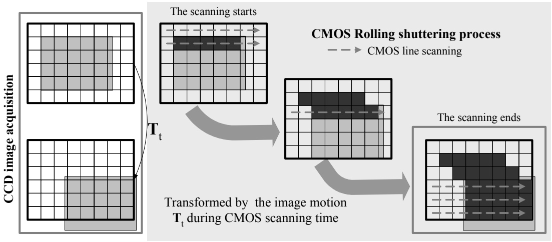

CCD image acquisition
$\mathbf{T}_t$
The scanning starts
CMOS Rolling shuttering process
CMOS line scanning
The scanning ends
Transformed by the image motion
$\mathbf{T}_t$ during CMOS scanning time

**Fig. 3 An image acquisition process in a CCD and a CMOS sensor**

Fig. 3 illustrates this problem in comparison with a CCD acquisition process. The diagram in its left side shows the process of a CCD image acquisition and its right side shows the time-flow illustration of a CMOS rolling shutter mechanism. Since a CCD sensor exposes all rows of pixels simultaneously and then stops the exposure to read out all pixel values for its image formation with an electronic global shutter, the captured image is the scene appearing at the time of exposure. But, in a CMOS sensor, while the sensor reads out the exposed row one by one to make an image, its other scanlines except for the current read-out row are still being exposed because there is no global shutter. This is called a *rolling shutter*. Thus, if a camera is moving during the rolling shutter time, then the captured scene is also being moved by its motion, which causes the deformation in a CMOS image. This deformation is not observed in a CCD image as shown in Fig. 3 and is always mingled with a camera motion that cannot be solved by previous DIS techniques without considering the rolling shutter mechanism.

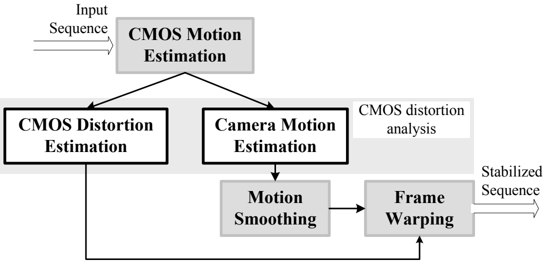


**Fig. 4. The block diagram of CMOS DIS algorithm**

Fig. 4 shows the basic block diagram of a CMOS DIS algorithm. *CMOS distortion analysis* carries out the process of decoupling a global image motion and a CMOS distortion based on a planar projective transformation between adjacent CMOS frames, and then the decoupled transformations are used to recover new frames of a stabilized video sequence. The CMOS distortion analysis in Fig. 4 is the important process in a CMOS video stabilization because it can solve CMOS distortion problem.

The concept of the CMOS distortion was, firstly, suggested by Hwang [2] based on a translational motion. But his approach was very restricted to a translational motion and the relation between a CMOS frame motion and its related global image motion was not clarified. In [5], they also used a translational motion model to remove CMOS artifacts by a line-shifting operation. Even though they tried to remove a CMOS distortion based on the rolling shutter mechanism, their approach was also restricted to a translational motion. Moreover, it was difficult to apply to more general global image motion like zooming and rotational motion because they did handle the distortion problem by a line-shifting operation. To approach this problem more mathematically, Im et al. [1] tried to estimate the CMOS distortion using a translational motion model based on a homography between CMOS frames but there was much room for improvement in more general image motion case.

In this paper, we propose a *CMOS distortion model* taking into account the rolling shutter mechanism in the case of *a zooming and translational motion*. The reminder of this paper is organized as follows: Section II covers the overall algorithm with the explanation of essential notations. In Section III, we propose the *CMOS distortion model* based on the rolling shutter mechanism and derive the CMOS distortion transformation to remove CMOS artifacts combined with a camera motion. Section IV explains the stabilization process to obtain an undistorted and stabilized CMOS frame sequence. In Section V, we show by experimental results that our distortion model can remove the rolling shutter distortion and also give a test result of an outdoor activity sequence taken by a commercial rolling shutter camcorder.

## II. Overall Procedure

This section covers the overall procedure of our proposed CMOS DIS algorithm and Fig. 5 shows a brief relationship between important processing parts. All notations and their relations will be previewed in this section.

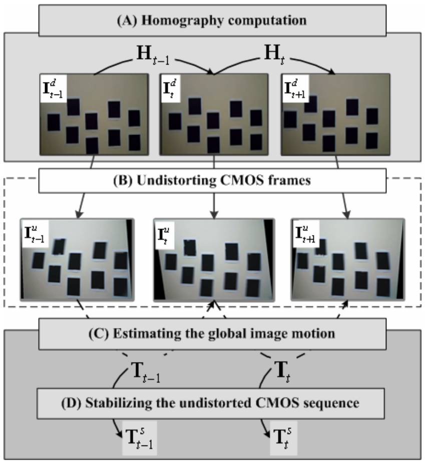


**Fig. 5. the overall CMOS DIS procedure**

### A. Homography computation

To obtain motion information from CMOS frames, we use the planar projective transformation between two frames called a homography. We estimate a CMOS image motion [1] by a CMOS homography $\mathbf{H}_t$ between adjacent CMOS frames,

$\mathrm{I}_t^d$ and $\mathrm{I}_{t+1}^d$ related by

$$\begin{pmatrix} \mathbf{x}_{t+1}^d \\ 1 \end{pmatrix} = \mathbf{H}_t \begin{pmatrix} \mathbf{x}_t^d \\ 1 \end{pmatrix}, \quad (1)$$

where $\mathbf{x}_t^d$ denotes a pixel position of an image $\mathrm{I}_t^d$ and $\mathbf{H}_t$ can be calculated from measurements of CMOS frames [3].

### *B. A global image motion and CMOS distortion transformation*

We estimate the global image motion $\mathbf{T}_t$ between undistorted frames, $\mathrm{I}_t^u$ and $\mathrm{I}_{t+1}^u$, and a CMOS distortion transformation $\mathbf{D}_t$ from the homography $\mathbf{H}_t$. $\mathbf{T}_t$ is a zooming and translational motion model which has the form of

$$\begin{pmatrix} \mathbf{x}_{t+1}^u \\ 1 \end{pmatrix} = \mathbf{T}_t \begin{pmatrix} \mathbf{x}_t^u \\ 1 \end{pmatrix} = \begin{pmatrix} a_t & 0 & c_t \\ 0 & a_t & d_t \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{x}_t^u \\ 1 \end{pmatrix}, \quad (2)$$

where $\mathbf{x}_t^u$ denotes a pixel position of an image $\mathrm{I}_t^u$ and the CMOS distortion transformation $\mathbf{D}_t$ relates $\mathrm{I}_t^d$ and $\mathrm{I}_t^u$ by

$$\begin{pmatrix} \mathbf{x}_t^d \\ 1 \end{pmatrix} = \mathbf{D}_t \begin{pmatrix} \mathbf{x}_t^u \\ 1 \end{pmatrix}. \quad (3)$$

### *C. Stabilizing an undistorted CMOS frame sequence*

By applying a smoothing filter to global image motions $\mathbf{T}_t$, we obtain stabilized global image motions $\mathbf{T}_t^s$ and make a stabilized undistorted CMOS frames $\mathrm{I}_t^s$. As each CMOS frame goes through the undistorting process in Fig. 4, there are no CMOS distortions in the new stabilized video frames.

## III. CMOS DISTORTION ANALYSIS

In this section, we will propose the CMOS distortion model based on the CMOS rolling shutter mechanism and explain the way of analyzing CMOS distortions in CMOS frames.

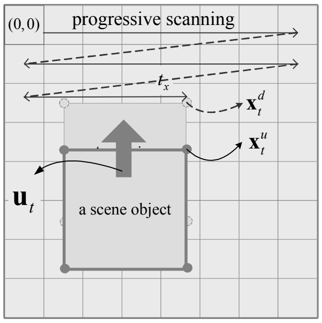


**Fig. 6. The CMOS pixel movement during a rolling shuttering time**

### *A. CMOS distortion model*

Fig. 6 shows how the feature points of a scene object move from their initial positions during a scanning time under a rolling shutter condition. Here, it is assumed that when a CMOS camera with an image resolution W×H is moving down, its captured scene object is moving up with its corresponding image motion $\mathbf{u}_t$ during one frame time.

Suppose that for a frame whose scanning starts at $t = 0$, an initial position of one feature $\mathbf{x}_t^u$ moves with this motion $\mathbf{u}_t$ during one frame time. After time $t_x = x_t^d / WH + y_t^d / H$ which is the elapsed time to a pixel position $\mathbf{x}_t^d = (x_t^d, y_t^d)^T$ from the image origin, the undistorted position $\mathbf{x}_t^u$ has been moved to the CMOS distorted position $\mathbf{x}_t^d$ satisfying the CMOS distortion model of

$$\mathbf{x}_t^d = \mathbf{x}_t^u + \int_{t=0}^{t_x} \mathbf{u}_t dt. \quad (4)$$

Note that the CMOS distorted position $\mathbf{x}_t^d$ comes to be exposed at a different position from its undistorted position $\mathbf{x}_t^u$ by $\int_{t=0}^{t_x} \mathbf{u}_t dt$ where $t_x$ can be approximated by $t_x \approx y_t^d / H$. Assuming that the global image motion $\mathbf{T}_t$ (2) does not change during the rolling shutter time, $\mathbf{x}_t^d$ is thought to be moved by the local motion $\mathbf{u}_t(\mathbf{x}_t^u)$ at $\mathbf{x}_t^u$ during the elapsed time $t_x$ and then (4) can be approximated to

$$\mathbf{x}_t^d = \mathbf{x}_t^u + \mathbf{u}_t(\mathbf{x}_t^u) \frac{y_t^d}{H}, \quad (5)$$

where $\mathbf{u}_t(\mathbf{x}_t^u)$ has the form of

$$\mathbf{u}_t(\mathbf{x}_t^u) = \begin{pmatrix} a_t & 0 \\ 0 & a_t \end{pmatrix} \mathbf{x}_t^u + \begin{pmatrix} c_t \\ d_t \end{pmatrix} - \mathbf{x}_t^u = \begin{pmatrix} (a_t - 1)x_t^u + c_t \\ (a_t - 1)y_t^u + d_t \end{pmatrix}. \quad (6)$$

Substituting $\mathbf{u}_t(\mathbf{x}_t^u)$ in (5) by (6) we obtain the mapping relation between $\mathrm{I}_t^u$ and $\mathrm{I}_t^d$ as follows

$$x_t^u = \frac{Hx_t^d - c_t y_t^d}{H + (a_t - 1) y_t^d}$$
$$y_t^u = \frac{(H - d_t) y_t^d}{H + (a_t - 1) y_t^d}. \quad (7)$$

The distortion transformation $\mathbf{D}_t$ between $\mathrm{I}_t^d$ and $\mathrm{I}_t^u$ can be derived from (7), which can be represented by the projective transformation form of

$$\begin{pmatrix} \mathbf{x}_t^u \\ 1 \end{pmatrix} = \left(\mathbf{D}_t\right)^{-1} \begin{pmatrix} \mathbf{x}_t^d \\ 1 \end{pmatrix} = \begin{pmatrix} H & -c_t & 0 \\ 0 & H-d_t & 0 \\ 0 & a_t-1 & H \end{pmatrix} \begin{pmatrix} \mathbf{x}_t^d \\ 1 \end{pmatrix}. \quad (8)$$

Using (7), CMOS distortions shown in Fig. 2 can be explained for typical camera motions.

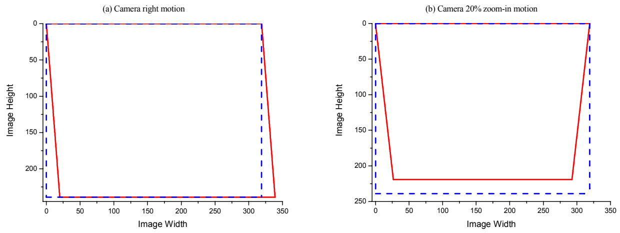


**Fig. 7. The simulation results of a CMOS frame $\mathrm{I}_t^d$ and its undistorted frame $\mathrm{I}_t^u$ by (7): (a) a camera right motion ($c_t = -20$) and (b) a camera 20% zoom-in motion.**

Fig. 7 illustrates a mapping relation between $\mathrm{I}_t^d$ and $\mathrm{I}_t^u$ for a camera motion of right (Fig. 7(a)) and zoom-in motion cases (Fig. 7(b)). A dotted line area (a rectangle region) is a CMOS image $\mathrm{I}_t^d$ with 320×240 and a solid line area is its undistorted image $\mathrm{I}_t^u$ mapped by (7). Fig. 7 means that the dotted rectangle frame $\mathrm{I}_t^d$ should be warped to the solid line area to obtain an undistorted frame $\mathrm{I}_t^u$. For example, since the solid line area is slanted to left in Fig. 7(a), a CMOS image has a right-skewed distortion. In a zoom-in case of Fig. 7(b), the zoom scale 20% makes the warping area a trapezoidal shape but such an abrupt change of the scale usually does not occur in real situation.

### B. *A homography based CMOS motion estimation*

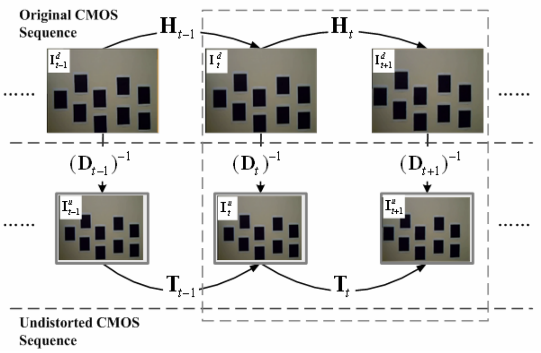


**Fig. 8. The relation between a CMOS homography, two distortions and a global motion.**

Fig. 8 shows the relation between a CMOS homography $\mathbf{H}_t$, global motion $\mathbf{T}_t$ and CMOS distortions, $\mathbf{D}_t$ and $\mathbf{D}_{t+1}$, which can be given by

$$\begin{pmatrix} \mathbf{x}_{t+1}^d \\ 1 \end{pmatrix} = \mathbf{H}_t \begin{pmatrix} \mathbf{x}_t^d \\ 1 \end{pmatrix} = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{x}_t^d \\ 1 \end{pmatrix} \quad (9)$$

$$= \mathbf{D}_{t+1}\mathbf{T}_t(\mathbf{D}_t)^{-1} \begin{pmatrix} \mathbf{x}_t^d \\ 1 \end{pmatrix}.$$

As for the homography computation method, there have been numerous works [3], [12] and in this paper we used a fast technique which was used in the mosaic algorithm [3]. From (9), the CMOS homography $\mathbf{H}_t$ is equal to the following parameterized projective transformation by substituting (2) and (8) for (9):

$$\mathbf{H}_t \approx \mathbf{D}_{t+1}\mathbf{T}_t(\mathbf{D}_t)^{-1} =$$

$$\begin{pmatrix} a_t & -\frac{a_t c_t}{H} + \frac{a_t c_{t+1}}{H} \cdot \frac{H-d_t}{H-d_{t+1}} + \frac{(a_t-1)c_t}{H} + \frac{(a_t-1)d_t c_{t+1}}{H(H-d_{t+1})} & c_t + \frac{d_t c_{t+1}}{H-d_{t+1}} \\ 0 & a_t \frac{H-d_t}{H-d_{t+1}} + \frac{(a_t-1)d_t}{H-d_{t+1}} & \frac{Hd_t}{H-d_{t+1}} \\ 0 & \frac{a_t\left(1-a_{t+1}\right)}{H} \cdot \frac{H-d_t}{H-d_{t+1}} + \frac{a_t-1}{H} + \frac{d_t(a_t-1)(1-a_{t+1})}{H(H-d_{t+1})} & 1 + \frac{d_t\left(1-a_{t+1}\right)}{H-d_{t+1}} \end{pmatrix}. \quad (10)$$

To find the solutions of the six motion parameters ($a_t$, $c_t$ and $d_t$ are called *Time-t* parameters, and $a_{t+1}$, $c_{t+1}$ and $d_{t+1}$, *Time-t+1* parameters) we use the fact that the CMOS homography parameters $h_1$ □ $h_8$ in (9) are equal to each corresponding element in the matrix of (10) up to scale. Given $h_1$ □ $h_8$ computed between $\mathrm{I}_t^d$ and $\mathrm{I}_{t+1}^d$, we can obtain the solutions for the parameters, $a_t$, $c_t$, $d_t$, $a_{t+1}$, $c_{t+1}$ and $d_{t+1}$ because the number of the motion parameters and the number of equations in (10) are equal and their solutions are given by

$$a_t = \frac{h_1(Hh_5 + h_6)}{H(h_5 - h_6h_8) + h_1h_6} \quad (11)$$
$$c_t = \frac{H(h_3h_5 - h_2h_6)}{H(h_5 - h_6h_8) + h_1h_6}, \quad d_t = \frac{Hh_1h_6}{H(h_5 - h_6h_8) + h_1h_6},$$

$$a_{t+1} = 1 - \frac{1}{h_5}\left(Hh_8 - \frac{Hh_5(h_1-1) + h_6h_8}{Hh_5 + h_6}\right) \quad (12)$$
$$c_{t+1} = \frac{(Hh_2 + h_3)H}{Hh_5 + h_6}, \quad d_{t+1} = \frac{(h_5 - h_1)H^2 + Hh_6}{Hh_5 + h_6}.$$

Thus, we directly compute *Time-t* and *Time-t+1* parameters (11)-(12) using the estimated CMOS homography $\mathbf{H}_t$ which enable to compute CMOS distortion transformations $\mathbf{D}_t$, $\mathbf{D}_{t+1}$ and a global image motion $\mathbf{T}_t$.

### C. *Obtaining undistorted frames*

For each homography, we obtain two distortion transformations, *Time-t* $\mathbf{D}_t$ and *Time-t+1* $\mathbf{D}_{t+1}$ by using the motion parameters computed from (11)-(12). This means that for each CMOS frame $\mathbf{I}_t^d$, we have two choices for $\mathbf{D}_t$ s; *Time-t+1* $\mathbf{D}_t$ from $\mathbf{H}_{t-1}$ and *Time-t* $\mathbf{D}_t$ from $\mathbf{H}_t$. Then, we should decide on which one to choose to make an undistorted frame $\mathbf{I}_t^u$ from $\mathbf{I}_t^d$. Mathematically, the two parameter sets of these distortion transformations are the same under the uniform velocity motion of a camera because the proposed CMOS distortion model is based on this motion assumption. Thus, if the camera is moving abruptly with an acceleration motion, then *Time-t+1* parameters of the previous homography are not equal to *Time-t* parameters of the current homography. To give more practical reasoning on the parameter selection, we will show examples for simple motion cases which cause typical CMOS distortions.

For a horizontal motion case, since the vertical lines in CMOS frames are slanted left or right, its homography parameters in (9) computed between the two adjacent CMOS frames would have $h_1 \approx 1$, $h_2 \approx 0 \sim 0.08$, $h_5 \approx 1$ and $h_6 = 0$. Since $h_2$ reflects the degree of the x-directional distortion between two adjacent CMOS frames, the value of $h_2$ runs from 0 to 0.08 experimentally in this motion. Using these values for (11) and (12) we can achieve *Time-t* parameters, $a_t \approx h_1$, $c_t \approx h_3$, $d_t = 0$, and *Time-t+1* parameters, $a_{t+1} \approx 1$, $c_{t+1} \approx Hh_2 + h_3$, $d_{t+1} = 0$. Here, *Time-t* parameter $c_t$ is approximately close to the x-directional translation parameter $h_3$ of $\mathbf{H}_t$ which defines the current x-directional global translation motion between the two CMOS frames, while *Time-t+1* parameter $c_{t+1}$ has an additional term $Hh_2$ with $h_3$. Since $H$ is the number of rows of an image and its value is above several hundreds, $Hh_2$ can have a large erroneous value if the camera motion is not uniform. Thus, $c_t$ can be more reliably used to estimate the x-directional motion between frames than $c_{t+1}$. For a vertical motion case, we can also give the same reasoning. In this case, because the scene objects in an image are vertically squeezed or stretched, the homography parameters would have $h_1 \approx 1$, $h_2 = h_3 = 0$ and $h_5 \approx 0.92 \sim 1.07$. Since $h_5$ reflects the degree of the y-directional distortion, experimentally $h_5$ has the value around 1.0, which is different from the horizontal motion case. With these values, we have *Time-t* parameters, $a_t \approx h_1$, $c_t = 0$, $d_t \approx h_6$, and *Time-t+1* parameters, $a_{t+1} \approx 1$, $c_{t+1} = 0$, $d_{t+1} \approx (h_5 - h_1)H + h_6$ from (11) and (12). Here, *Time-t* $d_t$ is close to the y-directional translation parameter $h_6$ of $\mathbf{H}_t$ while *Time-t+1* $d_{t+1}$ has an additional term $(h_5 - h_1)H$ with $h_6$. Due to the large value $H$, $d_{t+1}$ can have a large erroneous values in the acceleration motion case even if $h_5 - h_1$ has a small value around $0.07 \sim -0.08$. However, $d_t$ represents the current y-directional global translation motion between the two CMOS frames.

From the two examples, *Time-t* parameters approximately follow the global motion parameters between two CMOS frames. *Time-t+1* parameters are influenced by the degree of distortion between CMOS frames because $h_2$ and $h_5$ are associated with the distortion between the frames. Thus, if the camera motion is not uniform, *Time-t+1* parameters can be influenced by $h_2$ and $h_5$ due to a large value $H$ but *Time-t* parameters always reflect the global image motion between the frames as explained in the examples. Therefore, we use these *Time-t* parameters more reliably to compute the CMOS distortion $\mathbf{D}_t$ for $\mathbf{I}_t^d$.

In the final step of the proposed CMOS distortion analysis, the undistorted frame sequence consists of undistorted frames $\mathbf{I}_t^u$ obtained by applying *Time-t* distortion $\mathbf{D}_t$ to CMOS frames $\mathbf{I}_t^d$ and the estimated global motion $\mathbf{T}_t$ with *Time-t* parameters (11).

## IV. Digital Image Stabilization Algorithm

In this section, we will explain our stabilization process: smoothing and making a new stabilized frame sequence from an undistorted CMOS one.

Now we have obtained the undistorted CMOS frames $\mathbf{I}_t^u$ by using *Time-t* distortion transformation $\mathbf{D}_t$ and *Time-t* global image motions $\mathbf{T}_t$ after the CMOS distortion analysis. The next step is to obtain a stabilized motion by applying a low-pass filtering to an accumulative global transformation $\mathbf{U}_t$ of $\mathbf{T}_t$ by up to time *t*, $\mathbf{U}_t = \prod_{k=1}^{t-1} \mathbf{T}_{t-k}$, which means that the first frame $\mathbf{I}_1^u$ of an input sequence is set to the reference frame. The reason is that we want the increments of parameters in $\mathbf{U}_t$ to change smoothly from the first frame because their increments indicate the global image transformation between adjacent frames [12]. The stabilized accumulative global transformation $\mathbf{S}_t$ is made by applying a smoothing filter to the original sequence of $\mathbf{U}_t$. For its smoothing filter, we used a simple 2nd order IIR low pass filter [11]. Since an IIR filter has a recursive form which uses previous filtered data, it is applicable to a video processing and is easy to design the filter to show a good performance. For the final step of the CMOS stabilization, the motion-compensated CMOS frame $\mathbf{I}_t^s$ is obtained by warping $\mathbf{I}_t^u$ with its warping transformation $\mathbf{W}_t$ that can be written by

$$\begin{pmatrix} \mathbf{x}_t^s \\ 1 \end{pmatrix} = \mathbf{W}_t \begin{pmatrix} \mathbf{x}_t^u \\ 1 \end{pmatrix} = \mathbf{S}_t (\mathbf{U}_t)^{-1} \begin{pmatrix} \mathbf{x}_t^u \\ 1 \end{pmatrix}, \qquad (13)$$

where $\mathbf{W}_1 = \mathbf{I}_{3\times 3}$ is the identity matrix and $\mathbf{x}_t^s$ denotes a pixel position of an image $\mathbf{I}_t^s$. Then, we achieve the stabilized CMOS frame sequence without CMOS distortions.

For the stabilization test, we will compare the stabilized global image motion $\mathbf{T}_t^s$ between motion-compensated CMOS frames, $\mathbf{I}_t^s$ and $\mathbf{I}_{t+1}^s$, with the original global image motion $\mathbf{T}_t$. $\mathbf{T}_t^s$ is related by the warping transformations and the global image motion as follows

$$\begin{pmatrix} \mathbf{x}_{t+1}^s \\ 1 \end{pmatrix} = \mathbf{T}_t^s \begin{pmatrix} \mathbf{x}_t^s \\ 1 \end{pmatrix} = \mathbf{W}_{t+1}\mathbf{T}_t(\mathbf{W}_t)^{-1}\begin{pmatrix} \mathbf{x}_t^s \\ 1 \end{pmatrix}, \qquad (14)$$

and their comparison result will be shown in the following section.

## V. Experiments

In the experiments, we used a USB PC webcam and a commercial CMOS digital hand-held camcorder. The PC webcam uses a color 330K-pixel CMOS image sensor with a video resolution VGA 640×480 at 15 fps, and the commercial model uses a 3.2M-pixel CMOS image sensor with a video resolution VGA 640×480 at 30 fps. For a CMOS distortion test, we used the webcam because CMOS characteristics can be easily observable. For an outdoor video test, we used the commercial model because it shows a good image quality as a commercial CMOS camcorder.

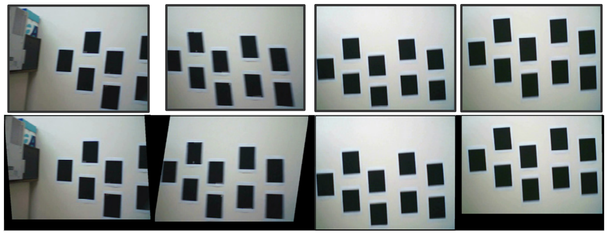


**Fig. 9. Examples of the proposed CMOS distortion analysis results for the right, left, down and up motions: (Top) the original CMOS frames and (Bottom) their corresponding undistorted frames.**

In Fig. 9, we tested our CMOS distortion model for four kinds of a simple camera motion. The top row shows the original CMOS frames which have CMOS distortions and in the bottom are their undistorted frames obtained by the distortion transformations (8). In the horizontal motion case, the original CMOS frames include left or right slanted rectangle patterns and in the vertical motion case, the same patterns appear stretched or squeezed. But, in our undistorted frame results those patterns become vertically upright and squeezed or stretched patterns are restored to their regular heights.

In Fig. 10, we quantify the uprightness and the height regularity of the same rectangle patterns by measuring their slant angle and height (Fig. 10(a)) in original CMOS frames $\mathbf{I}_t^d$ and our undistorted frames $\mathbf{I}_t^u$ respectively. The results are plotted in Fig. 10(b)-(e) for each camera motion case where the dashed lines show the graphs for $\mathbf{I}_t^d$ and the solid lines show the graphs for $\mathbf{I}_t^u$ with the two average values of all the patterns in each frame. Fig. 10(b)-(c) show a fast left and right motion case of a camera. In Fig. 10(c), the average slant angle graph plotted by dashed line shows a sinusoidal shape but its undistorted solid line graph becomes constant around the upright angle of 90°. Fig. 10(d)-(e) is for up and down motion case. In Fig. 10(d), the average height plotted by the dashed line has a large fluctuation indicating the large variations in the height of the patterns. But its undistorted solid line graph shows that its large fluctuation has been removed so that their average height appears regularly. From these results, the proposed CMOS distortion analysis procedure can remove CMOS distortions for each camera motion case. Thus, we can confirm that our algorithm can properly handle CMOS distortion problem coupled with its camera motion.

Fig. 11 shows the comparison results of a previous CCD-based DIS algorithm and our proposed CMOS DIS algorithm. Previous DIS techniques also stabilized a visually unstable CMOS video by using image motions directly estimated from frames. But to demonstrate the difference, we used a plug-in type shareware version of a video editing program, which shows a good stabilization result in [4]. When we tested on a CMOS video sequence with this software, CMOS distortions, as expected, still remained in its stabilization video. Here, the top row shows left or right slanted distortions in CMOS frames and its middle row shows the results processed by [4].

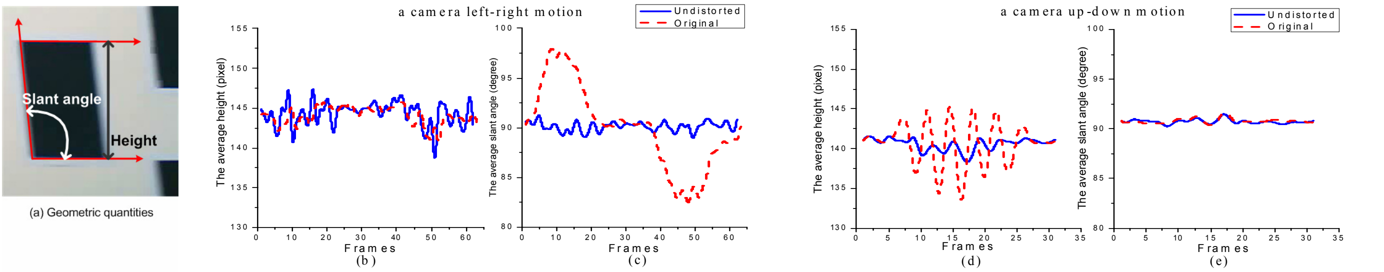


**Fig. 10. The geometric quantitative comparison between original and undistorted frames: (a) two geometric measurements of a rectangle for a left-right camera motion case (b)-(c), and for an up-down camera motion case (d)-(e). For each case, the average height (unit: pixel) and slant angle (unit: degree) are plotted for original frames (dashed line) and our undistorted frames (solid line) respectively.**

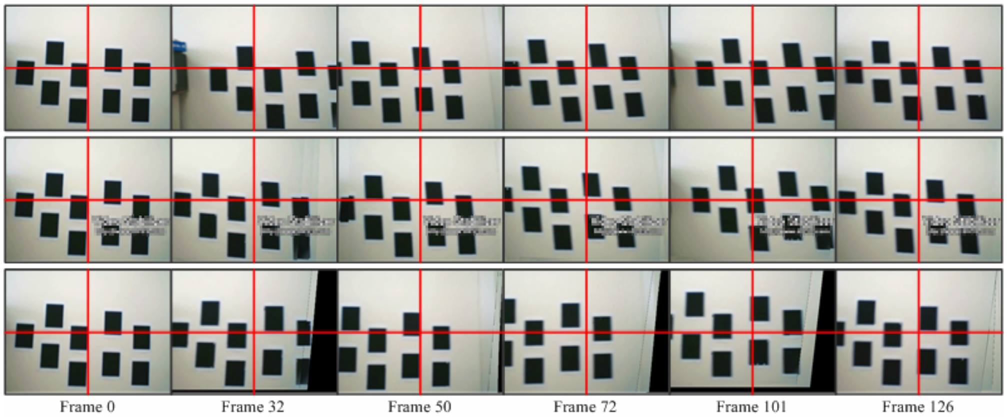


**Fig. 11. The comparison of a CCD DIS and our proposed CMOS DIS algorithm for a CMOS video taken by a CMOS webcam with a fast left and right motion: (Top) an original CMOS video, its stabilized results obtained by (Middle) a CCD-based DIS method and (Bottom) our proposed CMOS DIS algorithm.**

When compared to the original sequence, there are no differences except that its motion is smoothed. But our results in the bottom row show that the same patterns appear more uprightly than those in its original CMOS frames, which has been confirmed by our quantitative comparison results plotted by solid line in Fig. 10.

In Fig. 13, we tested our algorithm with an outdoor CMOS video taken by the commercial CMOS model hand-held digital camcorder. Since the CMOS video clip of Fig. 13 was recorded while going downstairs, it has large movements in y-direction which is shown in Fig. 12.

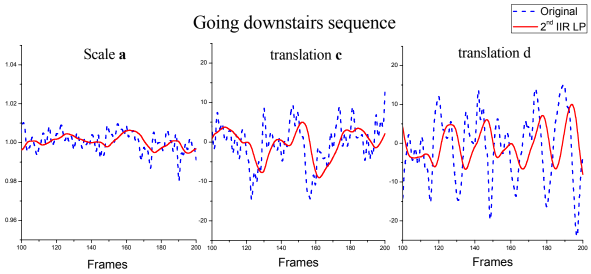


**Fig. 12. The stabilization result of “Going downstairs” video clip: the original global image motion (2) and its smoothed one (14) by 2$^{nd}$ IIR Low-pass filtering are plotted by dotted line and solid line respectively. From the left are three graphs for $a_t$, $c_t$ and $d_t$.**

Here, the original motion $\mathbf{T}_t$ in (2) is plotted by dotted line and the smoothed motion $\mathbf{T}_t^s$ in (14) plotted by solid line was filtered by 2$^{nd}$ order IIR low-pass filter. Even though this commercial model exhibits less CMOS distortion than a CMOS webcam, the rolling shutter camcorder cannot keep away from its CMOS distortion when seeing its video clips.

## VI. Conclusion

In this paper, we proposed a novel approach to more general CMOS digital image stabilization. By incorporating the rolling shutter mechanism into the CMOS distortion model, our algorithm could achieve the removal of CMOS distortions as well as the stabilization of a CMOS video sequence for a zooming and translational motion. Through our experimental results, we showed that the proposed CMOS distortion analysis gave a way of handling CMOS distortion more properly, which was not considered in previous DIS algorithms. However, because it is difficult to apply more general global motion model under the proposed homography based framework, we are considering the method of applying the full affine global image motion as future work.

## References

[1] Jeong-A Im, Dae-Wong Kim, and Ki-sang Hong, “Digital video stabilization algorithm for CMOS image sensor,” IEEE Proc. Int’l Conf. on Image Processing., Oct. 2006.

[2] J. H. Hwang, “CMOS image sensor distortion compensation method,” Korea Patent 10-0092224, 2003.

[3] Dae-Wong Kim and Ki-Sang Hong, “Real-time mosaic using sequential graph,” Journal of Electronic Imaging, vol. 15, no. 2, pp. 0230051-02300516, Apr.-Jun. 2006.

[4] DiGiStudio Team, “DiGiStudio Video stabilization 1.2D,” the plug-in shareware version for commercial video editing programs in http://www.dv99.com.

[5] C.-K. Liang and H. H. Chen, “Rolling shutter distortion correction,” SPIE Proc. Visual Communications and Image Processing, Beijing, China, Jul. 2005.

[6] Haruhisa Okuda, Manabu Hashimoto, Kazuhiko Sumi, and Shun’ichi Kaneko, “Optimal motion estimation algorithm for fast and robust digital image stabilization,” IEEE Trans. Consumer Electron., vol. 52, no. 1, pp. 276-280, Feb. 2006.

[7] Lidong Xu, and Xinggang Lin, “Digital image stabilization based on circular block matching,” IEEE Trans. Consumer Electron., vol. 52, no. 2, pp. 566-574, May 2006.

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

[9] K. J. Hwan, “Adaptive digital image stabilization using bit-plane matching”, Comm. & Multimedia Electronics Workshop, pp.47-52, 1998.

[10] Jyh-Yeong Chang, Wen-Feng Hu, Mu-Huo Cheng, and Bo-Sen Chang, “Digital image translational and rotational motion stabilization using optical flow technique,” IEEE Trans. Consumer Electron., vol. 48, no. 1, Feb. 2002.

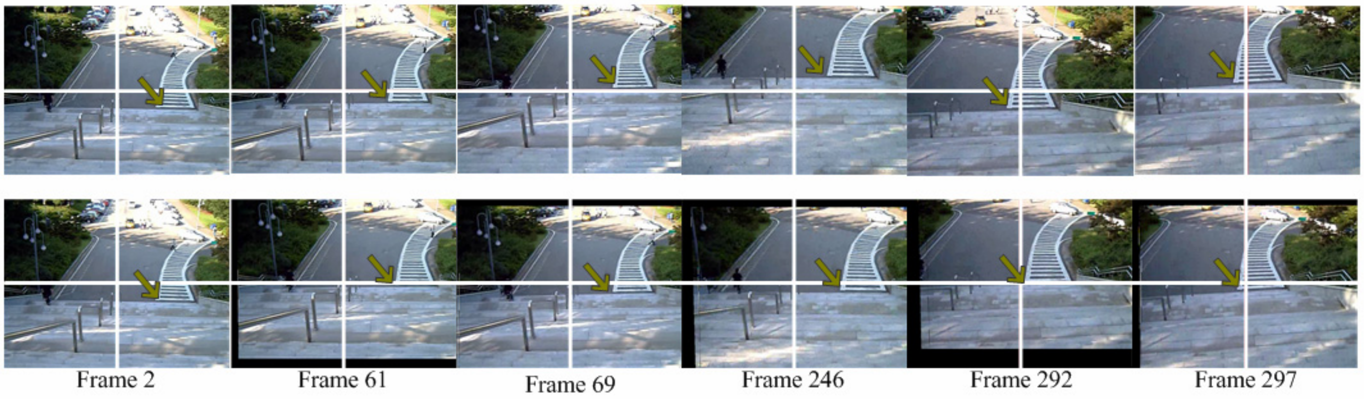


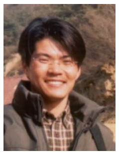
**Fig. 13. "Going downstairs" video clip: The first row is an original CMOS video clip and its second is our stabilization results where the frame numbers are indicated for the two cases. See the crosswalk lines through original frames and compare them with our results.**

[11] A. J. Crawford, H. Denman, F. Kelly, F. Pitie, and A. C. Kokaram, "Gradient based dominant motion estimation with integral projections for real time video stabilization," IEEE Proc. Int'l Conf. on Image Processing, vol. 5, pp. 3371-3374, Oct. 2004.

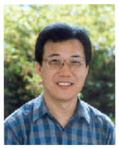
[12] Yasuyuki Matsushita, Eyal Ofek, Weina Ge, Xiaoou Tang, and Heung-Yeung Shum, "Full-frame video stabilization with motion inpainting," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 28, no. 7, pp. 1150-1163, Jul. 2006.

[13] Andrew Litvin, Janusz Konrad, and William C. Karl, "Probabilistic video stabilization using Kalman filtering and mosaicking," SPIE Proc. Electronic Imaging, vol. 5022, pp. 663-674, Jan. 2003.

**Won-Ho Cho** received the B.S. degree from Electronic and Electrical Engineering, KAIST, S. Korea in 1996 and received the M.S. degree from Electronic and Electrical Engineering, POSTECH, S. Korea in 2001. In 1998, he worked as a visiting researcher in ETRI, S. Korea and he is currently a PH.D candidate in Electronic and Electrical Engineering in POSTECH, S. Korea. His research interests include multi-object tracking, high dynamic range imaging, contour based segmentation and video analysis.



Dae-Woong Kim received the B.S. degree from Electronic Engineering, Honam University, S. Korea in 1999, the M.S. degree and PhD degree in Electronic and Electrical Engineering, POSTECH in 2001 and 2007 respectively. In 2002, he worked as a visiting scientist in Microsoft Research Asia at Beijing and he is currently with Samsung Electronics Corp., S. Korea. His research interests include image registration, high dynamic range imaging and machine learning.



**Ki-Sang Hong** received the B.S. degree in Electronic Engineering from Seoul National University, S. Korea in 1977, the M.S. degree in Electrical and Electronic Engineering in 1979, and the PhD degree in 1984 from KAIST, S. Korea. From 1984 to 1986 he was a researcher with Korea Atomic Energy Research Institute and in 1986 he joined POSTECH, Korea, where he is currently a professor with the Division of Electrical and Computer Engineering. From 1988 to 1989 he was a visiting professor with Robotics Institute at Carnegie Mellon University, Pittsburgh, Pennsylvania. His current research interests include computer vision, augmented reality, pattern recognition, and color image processing.