AUTOMATIC VIDEO DESHEARING FOR SKEW SEQUENCES CAPTURED BY ROLLING SHUTTER CAMERAS

Dũng Trung Võ, Surapong Lertrattanapanich and Yeong-Taeg Kim

Digital Media Solutions Lab, Samsung Information Systems America Inc.
(Samsung Electronics US R&D Center), Irvine, CA 92612, USA

ABSTRACT

A novel automatic video deshearing is proposed to reduce the skew artifacts (jelly artifacts) in sequences captured by CMOS censor cameras. The paper first considers the principles of skew artifacts in rolling shutter cameras. Then a deshearing algorithm with high accuracy is discussed with automatic skew artifact detection. A video completion step is also implemented to reconstruct unavailable pixels in the new grid of the desheared frame. Simulation results show that the proposed method effectively reduces the skew artifacts for different sequences captured by rolling shutter cameras.

Index Terms— video deshearing, rolling shutter cameras, skew artifacts, video completion.

1. INTRODUCTION

CMOS censors are popularly implemented in hand-held cellphone cameras and DSLR cameras due to their lower cost than CCD censors. Different than whole frame sampling in CCD censors, row by row sampling is utilized in CMOS censors. They are so called rolling shutter cameras. This sampling scheme helps reducing buffer memory on the censor as well as permitting on chip auto-focus and white balance [1]. But due to row by row sampling, there will be a time elapse between different row sampling. If the motion is fast, straight lines and structured objects located in different rows will be bent. This causes shearing artifacts [2]. Fig. 1 shows an example of shearing artifacts in one frame of the sequence recorded using a CMOS rolling shutter camera. The buildings in the frame are skewed to the right due to fast right camera panning. In case the sequences are captured by CMOS censors while moving, the distortion may exist in different directions and results in wobble artifacts [3]. These artifacts should be removed to enhance the quality of captured sequences.

Not a lot of works have been mentioned on this topic. [4] [5] first built up a mathematical model for shearing artifacts and based on that to develop the deshearing algorithm. Both require a high complex model estimation with 3D correspondences. Also starting with a mathematical model, [2] interpolated motion of each row using global motion vector. Although proved to effectively remove the skew artifacts, it requires an user-aids step to indicate the matching pixels between normal frame and skew frame. Without consideration of the row by row scanning scheme, [6] utilizes a digital image stabilization to remove the distortion. In another approach, [3] assumed that motion of each pixel is a combination of low-frequency independent motion and a high-frequency camera jitter. The shearing is reduced by frame-based re-rendering using high-frequency camera motion. This method is not automatic and requires a calibration of the sampling time between rows.

Fig. 1. An example of shearing artifact.

When the motion is panning and the captured frame including vertical edges such as buildings, most of the previous methods do not provide an automatic solution to correctly deshear bent edges into vertical edges. [3] and [6] tries to keep the object shape consistent over frames but they do not guarantee to achieve the vertical edges. Vertical edges can be achieved by using methods in [1], [2] and [7] but they require calibration of read out time and oscillation frequency [1] or an user-aids step as in [2] or camera identification [7]. This paper proposes an automatic deshearing method to correct skew edges. The algorithm first estimates the skewing parameter and camera motion speed then uses these parameters to correct skew artifacts.

The paper is organized as follows. Section 2 analyses the shearing artifacts in rolling shutter cameras and proposes a robust deshearing procedure based on the skew parameter and global motion vector. Estimations for these parameters are discussed in section 3. When desheared, pixels at integer coordinates of the shearing frame may be shifted to a non-integer coordinate in the desheared frame. Another issue is

978-1-4577-1302-6/11/$26.00 ©2011 IEEE

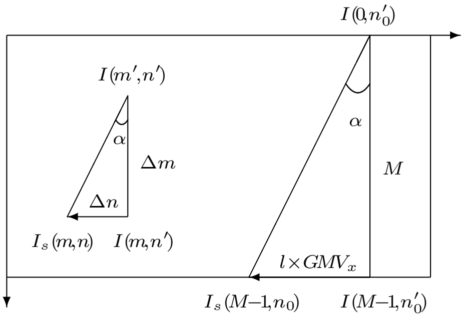


Fig. 2. Shearing angle detection.

that some pixels in the integer grid of the desheared frame may not have value during deshearing procedure. To resolve these issues, section 4 proposes an additional video completion step to reconstruct the missing pixel values. Simulation results are shown in section 5. Finally, section 6 gives the conclusion remarks and mentions future works.

2. SHEARING ARTIFACT ANALYSIS

This section investigates shearing artifact principles and discusses the proposed deshearing algorithm. Fig. 2 visualizes the shearing artifact principle of a vertical line. Assume that I is a $M \times N$ original frame without artifacts and $I(m', n')$ and $I(m, n')$ are two pixels in a vertical line on that frame. If the row including pixel $I(m', n')$ is read at time t, the row including pixel $I(m, n')$ will be read at time $t{+}\Delta t$ because of the row by row sampling scheme. Due to camera panning, the original pixel at (m, n') has moved to (m, n). The vertical line between $I(m', n')$ and $I(m, n')$ is bent to the diagonal line between $I(m', n')$ and $I_s(m, n)$ where $I_s(m, n)$ is the skew pixel of $I(m, n')$. Because analysis for skew artifacts resulted from horizontal and vertical panning are similar, so only artifacts from horizontal motion is considered in this section.

In video recording, the total time to sample one frame includes the reading time from the first row to the last row and the idle time between reading the last row of that frame and the first row of its next frame. If the idle time is insignificant comparing to the time elapse to read a frame, then the shearing artifacts proportionally depend on the panning speed of the camera. In that case, the skew distortion of the whole vertical line between pixels at $(0, n'_0)$ and $(M-1, n'_0)$ linearly depends on the panning speed of the whole frame. That means

$$n_0 - n'_0 = l \times GMV_x(t{+}1) \tag{1}$$

where $GMV_x(t{+}1)$ is the coordinate of global motion vector between frame t and frame $t{+}1$ in the horizontal direction and l is a constant scaling factor. For smaller vertical lines such as the one between two pixels $I(m', n')$ and $I(m, n')$ in Fig. 2, its skew distortion $\Delta n = (n{-}n')$ is smaller than $l \times GMV_x(t{+}1)$. Because the reading time for each row is equal, so the skew distortion is proportional to the row difference, thus

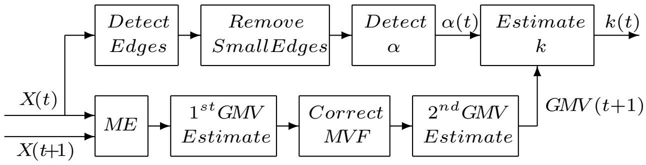


Fig. 3. Block diagram of $k(t)$ estimation.

$$\frac{l \times GMV_x(t{+}1)}{M} = \frac{\Delta n}{\Delta m} = tan\big(\alpha(t)\big) \tag{2}$$

where $\Delta m = (m{-}m')$ and $\alpha(t)$ is the skew angle of frame t. That leads to

$$\frac{l}{M} = \frac{tan\big(\alpha(t)\big)}{GMV_x(t{+}1)} = k_0. \tag{3}$$

Because l and M are fixed, so k_0 is constant. This equation agrees to the fact that higher $GMV_x(t{+}1)$ (faster panning speed) causes higher $\alpha(t)$ (more serious skew artifacts). The value of k_0 is varied for different camera settings such as camera types, frame rates or exposure time. If k_0 is known, then the deshearing process can be done by shifting pixel $I_s(m, n)$ back to location (m, n') of the desheared frame by a distortion distance Δn

$$\Delta n = k_0 \times \Delta m \times GMV_x(t{+}1). \tag{4}$$

The constant k_0 is estimated by using the average skew parameter $k(t)$ of a group of T frames

$$k_0 = \frac{1}{T}\sum_{t=0}^{T} k(t) = \frac{1}{T}\sum_{t=0}^{T}\left(\frac{tan\big(\alpha(t)\big)}{GMV_x(t{+}1)}\right). \tag{5}$$

Once k_0 has been determined, it can be used for the remaining frames in the sequences. If the sampling time for the whole frame is assumed to be at the middle row for minimal skew distortion [2], the pixel $I_s(m, n)$ of the skew frame is desheared to the new location (m, n') of desheared frame I_d

$$n' = n{-}\Delta n = n{-}k_0 \times (m{-}\frac{M}{2}) \times GMV_x(t{+}1). \tag{6}$$

3. SKEW ARTIFACT ESTIMATION

This section describes the procedure to estimate the skew parameter $k(t)$ in (5). As shown in Fig. 3, the whole process includes three estimations of global motion vector GMV, shearing angle $\alpha(t)$ and skew parameter $k(t)$.

3.1. GMV Estimation

The global motion is estimated based on the motion vector field (MVF) between the next frame $I(t{+}1, m, n)$ and the current frame $I(t, m, n)$. At first, a block-based motion estimation is implemented to form the motion vectors $MV(t{+}1, i, j)$

for every $(i,j)^{th}$ block. The motion is assumed to be caused by camera panning and the global motion vector $GMV(t+1)$ is considered as the most popular motion vector in the MVF

$$GMV(t+1) = \max_{MV(t+1,i,j)} \Big(P\big(MV(t+1,i,j)\big)\Big) \tag{7}$$

where $P(MV(t+1,i,j))$ is the number of vector $MV(t+1,i,j)$ occurrence. The GMV reliability is then calculated by

$$GMV_{rel}(t+1) = \frac{P\big(GMV(t+1)\big)}{P_0} \tag{8}$$

where P_0 is the total number of blocks in the frame. To avoid possible wrong motion vectors, an addition step is utilized to correct the MVF. The correction is based on the mismatched error. If the mismatched error using the estimated motion vector is greater than the mismatched error using $GMV(t+1)$, then the motion vector of that block is replaced by the $GMV(t+1)$. After the MVF is corrected, a second GMV estimation is implemented to recalculate the GMV with more accurate reliability. $GMV_x(t+1)$ is the horizontal coordinate of $GMV(t+1)$. The assumption of panning motion is validated if GMV_{rel} is higher than a predefined threshold.

If a higher accuracy is required for the GMV, then a further step can be applied to obtain sub-pixel GMV. The K most popular motion vectors $GMV(t+1,i)$ and its reliability $GMV_{rel}(t+1,i)$ are calculated based on (7) and (8). The sub-pixel GMV is formed by

$$GMV_{subpel}(t+1) = \frac{\sum_{i=1}^{K} GMV_{rel}(t+1,i) \times GMV(t+1,i)}{\sum_{i=1}^{K} GMV_{rel}(t+1,i)}. \tag{9}$$

3.2. Skew Angle and Skew Parameter Estimation

At first, an edge map is built using Sobel edge detector. Only significant edges are considered to increase the accuracy of skew artifact detection. Shearing angle is the angle which has the highest occurrence in the angle histogram of edge pixels. To reduce the risk of wrong detection, a low pass filter is applied to this histogram. Due to the motion direction, right panning results skew artifacts to the right, which leads to negative α value, and vice versa. So $GMV_x(t+1)$ and $\alpha(t)$ must have different signs. Furthermore, $GMV_x(t+1)$ should be large enough with high $GMV_{rel}(t+1)$ value to ensure a reliable skew angle $\alpha(t)$ detection. These conditions are then considered to validate $k(t)$ in k_0 estimation

$$k_0 = \frac{1}{T}\sum_{0}^{T} r(t) \times k(t) = \frac{1}{T}\sum_{0}^{T} \left(r(t) \times \frac{tan\big(\alpha(t)\big)}{GMV_x(t+1)} \right) \tag{10}$$

where $r(t)$ is the reliability of skew parameter $k(t)$

$$r(t) = \begin{cases} 1 & \text{if} \begin{cases} GMV_x(t+1) \geq GMV_{x,Th} \\ GMV_{rel}(t+1) \geq GMV_{rel,Th} \\ k(t) \leq 0 \end{cases} \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

where $GMV_{x,Th}$ and $GMV_{rel,Th}$ are predefined thresholds.

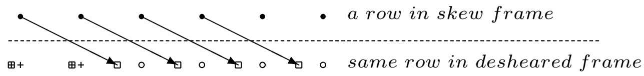


Fig. 4. An example of deshearing one row in the frame.

4. VIDEO COMPLETION

Once k_0 has been estimated based on the first T frames, the deshearing process can be implemented without the need of estimating $\alpha(t)$. In (6), the pixel at integer location (m,n) of skew frame is shifted to the new location at (m,n') of the desheared frame. This shifting brings up two issues. One issue comes from the possible non-integer co-ordinate of the new location (m,n'). The other issue is that there may be some pixels in the desheared frames which do not have any neighbor projected pixels from the skew frame. Fig. 4 illustrates an example in deshearing one row of the frame. The pixels at integer co-ordinate (pixels symbolized by solid circles) are shifted to the new non-integer co-ordinate in the desheared frame (pixels symbolized by squares). The target is to reconstruct pixels at integer grid points (pixel symbolized by blank circles or crosses). For pixels indicated by blank circles, their values can be found by interpolating the surrounding projected pixels (indicated by squares). The first issue is thus solved. For pixels which do not have surrounding shifted pixels (those symbolized by crosses), an additional step is needed to copy the pixels from surrounding frames to those locations. Assume the pixel which does not have value in the desheared frame is $I_d(m_m, n_m)$. At first, its matching pixels in the previous frame is found as $I_{pre}\big(m_m + GMV_x(t), n_m + GMV_y(t)\big)$ where $\big(GMV_x(t), GMV_y(t)\big)$ is the GMV of the current frame. The desheared pixel of the pixel at this location is $I_{pre}(m'_m, n'_m)$ where

$$\begin{cases} m'_m = m_m + GMV_x(t) \\ n'_m = n_m + GMV_y(t) - k_0 \times (m'_m - \frac{M}{2}) \times GMV_x(t). \end{cases} \tag{12}$$

Because (m'_m, n'_m) may be non-integer, so its nearest pixels $I_{pre}(\lfloor m'_m \rfloor, \lfloor n'_m \rfloor)$ in the previous frame is projected back to the current frame by a reverse procedure. Then an interpolation step is implemented to estimate the pixel value at integer grid point. The corresponding location of this pixel in the current frame is

$$\begin{cases} m_{m,r} = \lfloor m_m' \rfloor - GMV_x(t) \\ n_{m,r} = \lfloor n_m' \rfloor - GMV_y(t) + k_0 \times (\lfloor m_m' \rfloor - \frac{M}{2}) \times GMV_x(t). \end{cases} \tag{13}$$

From (12) and (13), if $GMV_x(t)$ accuracy is integer pixel, then $m_{m,r} = m_m$. The shifted pixel in the current frame is

$$I_d(m_m, n_{m,r}) = I_{pre}(m'_m, \lfloor n'_m \rfloor). \tag{14}$$

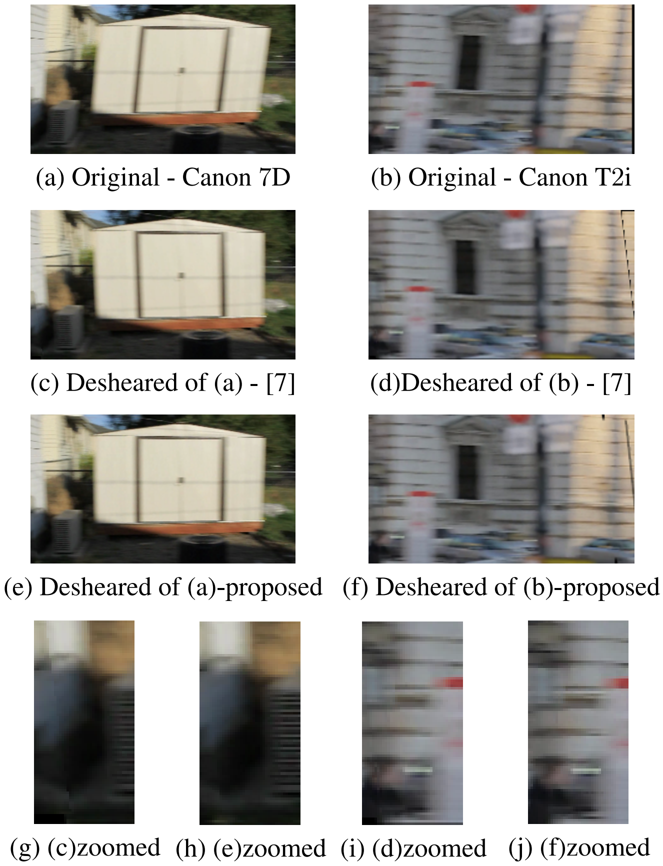
If the missing pixels' value cannot be copied from previous frame, they can be copied from other surrounding frames using the similar procedure. These pixels are indicated by squares with cross in Fig. 4. Together with other pixels at non-integer co-ordinates (pixels indicated by squares), they are used to interpolate the pixels at integer co-ordinates (pixels indicated by blank circles or crosses). Because deshearing pixels are in the same row with the shearing pixels, so one dimensional interpolation is applicable. This ensures a solution with low complexity and low line memory buffer. But more advanced interpolators such as [8] with sub-pixel motion estimation [9] can help achieving better result.

5. SIMULATION RESULTS

The video deshearing is simulated for uploaded Youtube videos captured by different CMOS cameras. $GMV_{rel,Th}$ are set to 0.1 while $GMV_{x,Th}$ is set to $3\%\times M$. Video completion uses row-based cubic interpolation. Simulation results for the 22^{nd} frame of a sequence captured by Canon 7D and the 61^{st} frame of a sequence captured by Canon T2i are shown in Fig. 5. For these sequences, the estimated skew parameters are $k_0 = -0.0030$ and -0.0046 for Canon 7D and Canon T2i, respectively. For comparison, the deshearing algorithm in [7] is also simulated. To get the best deshearing results in these simulations, the rolling shutter amount in [7] is tuned to 70% and 66% for Canon 7D and Canon T2i, respectively. Fig. 5 shows that both proposed method and method in [7] can effectively correct the skew artifacts. Note that [7] requires a tuning step for the rolling shutter amount. For the result of the frame captured by Canon T2i, there are artifacts near to the right border of the frame in Fig. 5(d) and Fig. 5(f) due to the black letter box in the original frame as shown in Fig. 5(b). The results using proposed method are also consistent over frames when played as video sequences, where the motion speed changes over frames. As seen in Fig. 5(h) and Fig. 5(j) with proposed method, border artifacts between the interpolated and copied areas are reduced comparing to the border artifacts in Fig. 5(g) and Fig. 5(i) with method from [7]. This validates the effectiveness of the video completion.

6. CONCLUSIONS

This paper proposes a novel method for video deshearing which can be applied without calibration. Based on the relation between the skewing angle and the camera panning speed, the deshearing algorithm uses the estimated global motion vector to determine the skew angle for deshearing. An additional video completion step is implemented to reconstruct the missing pixels during the deshearing. The proposed deshearing method helps reducing the skew artifact in different kinds of sequences. Future works focus on object-based video deshearing which can adapt deshearing objects with different speeds.



(a) Original - Canon 7D (b) Original - Canon T2i
(c) Desheared of (a) - [7] (d)Desheared of (b) - [7]
(e) Desheared of (a)-proposed (f) Desheared of (b)-proposed
(g) (c)zoomed (h) (e)zoomed (i) (d)zoomed (j) (f)zoomed

Fig. 5. Deshearing results of frames of sequences from Canon 7D and Canon T2i cameras.

7. REFERENCES

[1] P. Forssén and E. Ringaby, “Rectifying Rolling Shutter Video from Hand-held Devices,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 507 – 514, June 2010.

[2] C.K. Liang, L.W. Chang, and H. Chen, “Analysis and Compensation of Rolling Shutter Effect,” *IEEE Transactions on Image Processing*, vol. 17(8), pp. 13231330, August 2008.

[3] S. Baker, E. Bennett, S.B. Kang, and R. Szeliski, “Removing Rolling Shutter Wobble,” *IEEE Conference on Computer Vision and Pattern Recognition*, June 2010.

[4] C. Geyer, M. Meingast, and S. Sastry, “Geometric Models of Rolling Shutter Cameras,” *Omnidirectional Vision, Camera Networks and Non-classical Cameras*, pp. 12–19, October 2005.

[5] O. Ait-Aider and A. Bartoli, “Kinematics from Lines in a Single Rolling Shutter Image,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–6, June 2007.

[6] W.H. Cho and K.S. Kong, “Affine Motion Based CMOS Distortion Analysis and CMOS Digital Image Stabilization,” *IEEE Transactions on Consumer Electronics*, vol. 53(3), pp. 833841, August 2007.

[7] G. Thalin, “Deshaker,” *http://www.guthspot.se/video/deshaker.htm*, 2008.

[8] D.T. Vo, J. Sole, P. Yin, C. Gomila, and T.Q. Nguyen, “Selective Data Pruning-Based Compression Using High-Order Edge-Directed Interpolation,” *IEEE Transactions on Image Processing*, vol. 19, pp. 399–409, January 2010.

[9] Stanley H. Chan, Dung Vo, and Truong Q. Nguyen, “Subpixel Motion Estimation without Interpolation,” *IEEE Conf. on Acoustics, Speech & Signal Process.*, pp. 722–725, March 2010.