Calibration-Free Rolling Shutter Removal

Matthias Grundmann[1,2] Vivek Kwatra[1] Daniel Castro[2] Irfan Essa[1,2]
[1]Google Research, Mountain View, CA, USA [2]Georgia Institute of Technology, Atlanta, GA, USA
http://www.cc.gatech.edu/cpl/projects/rollingshutter

Abstract

We present a novel algorithm for efficient removal of rolling shutter distortions in uncalibrated streaming videos. Our proposed method is calibration free as it does not need any knowledge of the camera used, nor does it require calibration using specially recorded calibration sequences. Our algorithm can perform rolling shutter removal under varying focal lengths, as in videos from CMOS cameras equipped with an optical zoom. We evaluate our approach across a broad range of cameras and video sequences demonstrating robustness, scaleability, and repeatability. We also conducted a user study, which demonstrates preference for the output of our algorithm over other state-of-the art methods. Our algorithm is computationally efficient, easy to parallelize, and robust to challenging artifacts introduced by various cameras with differing technologies.

1. Introduction

Most current digital video cameras, from inexpensive cellphone cameras to high-end DSLRs, use active pixel sensors based on CMOS technology, as opposed to a charge coupled device (CCD). CMOS technology is appealing compared to CCDs due to its low power consumption, X-Y readout with optional skipping enabling on-chip exposure control during capture, and ease in manufacture as it shares the underlying process with almost all logic and microprocessors [13].

However, most cameras based on CMOS technology employ column parallel readout, also known as *electronic rolling shutter* [13]. Pixels within a row are read out simultaneously, but integration time is shifted row by row. A prior readout with optional pixel-skipping, usually shifted by half of a frame period is used to determine exposure time. As image rows are exposed and readout at different instances in time, electronic rolling shutter causes geometric image distortions ranging from shear, caused by low-frequency motions to wobble distortions caused by high frequency perturbations of the camera center. These wobble distortions are specifically noticeable in videos captured by cameras mounted on cars or helicopters and in videos captured by a walking person, which has motion spikes due to impact of the feet with the ground.

While these distortions are tolerable in still imaging, their temporal inconsistency is exaggerated for video. The magnitude of distortion primarily depends on the speed of the readout, *i.e.* readout time t_r w.r.t. the frame period T (alternatively, one might consider the inter-frame delay $T - t_r - t_e$, with t_e being the exposure time[13]). For this reason, high-end DSLRs with a faster readout time result in less distortion.

Current state of the art approaches require that this readout time t_r be determined *a-priori* [5, 8] in a controlled setting, or be calibrated from a video sequence recorded by the same camera prior to any corrections. This prevents the use of these algorithms in situations where only the video is available, without further knowledge of or access to the camera or the scene.

In this paper, we introduce a novel calibration-free algorithm for blind rolling shutter removal for video. Our contributions are:

- A novel mixture model of homographies parametrized by scanline blocks which faithfully models the inter-frame distortions caused by an electronic rolling shutter.
- An efficient estimation procedure robust to foreground motions leveraging regularization and iterative reweighted least squares.
- A thorough evaluation using various cameras and settings as well as a user study, which demonstrates general preference of our algorithm over others.
- A highly efficient solution, undistorting video at 5 - 10 fps on a single machine.

As rolling shutter distortions are caused by perturbations of the camera center, we perform joint rolling shutter removal and video stabilization. Specifically, we implemented the video stabilization method of Grundmann et al. [7], replacing their frame-pair registration with our homography mixtures as described in section 3.4. Examples of our results is shown in fig. 1.

2. Related work

Previous work on rolling shutter removal seeks to estimate parametric motion between two frames from feature matches while accounting for the time-varying manner of the capture process across rows (or blocks of rows).

Figure 1: Two examples rectified using our calibration free rolling shutter technique. Original frames on the left, our rectified result on the right. Our model accounts for frame global distortions such as skew (left example) as well as local wobble distortions which compress and stretch different parts of the frame (right example). Please see accompanying video.

Cho and Kong [4] employed global affine model estimation, which is used to derive a per pixel velocity (displacement per frame period). Rolling shutter correction is performed by multiplying the velocity with the actual capture duration between matches (expressed as the ratio of number of scanlines between matches to the number of scanlines per frame) yielding the actual displacement.

Liang et al. [11] use a per-row translation model obtained by interpolating frame global translations (*i.e.* one translational model per frame-pair) via Bezier curves. The translation is found as the peak in a 2D histogram of translation vectors obtained using block matching. Baker et al. [1] extend on this model by replacing Bezier interpolation with L1 regularization across scanlines, allowing for more general motions. They also account for independent motion, albeit optimization is costly in this case ($\sim$ 100s per frame).

Ringaby and Forssen [5, 6] extend upon Liang et al. [11] by interpolating 3D camera poses, specifically rotation matrices, across scanlines. In particular, they employ spherical linear interpolation resulting in a non-linear optimization problem.

The above mentioned rolling shutter removal techniques are limited in that a prior calibration is required to achieve good results. Baker et al. [1] assumes that the camera-dependent inter-frame delay is known a priori. While they demonstrate estimating this delay from a short clip recorded by the same camera, the clip is required to contain wobble distortion, which requires some degree of manual selection. Likewise, Ringaby and Forssen's [6] 3D calibration approach, requires considerable prior calibration. The intrinsic camera matrix is assumed to be known and constant during capture. More importantly, the inter-frame delay has to be determined prior to calibration, which is obtained by flashing a light source of known frequency. Lastly, the frame-rate is assumed to be known and remain constant during capture. In this respect, it should be noted that modern cell phone cameras employ dynamic frame-rates, *e.g.* the iPhone4 varies the frame rate from 24 fps in low-light settings to 30 fps if the scene is well lit [10].

Current video stabilization approaches, such as Liu et al. [12] treat rolling shutter distortions as noise and do not model it specifically. Similar, Grundmann et al. [7] model rolling shutter distortions via frame global homographies (*i.e.* one homography per frame-pair) and do not account for the time-varying nature of the capture process.

Most recently, the use of dedicated hardware was proposed to replace feature tracking within the rolling shutter framework of Ringaby and Forssen's [6]. In particular, Hanning et al. [8] and Karpenko et al. [10] simultaneously proposed to measure the camera rotations from gyroscopes. In addition to the inter-frame delay, both approaches require prior offline calibration of camera and gyroscope, which is performed per camera from a short video segment of a *planar* scene using high quality SIFT matches [10] or KLT feature tracks [8]. Our proposed algorithm does not require any such hardware nor any specific calibration, and can be applied to any video.

3. Calibration-free rolling shutter removal

We perform rolling shutter removal without the need for prior calibration by expressing the rolling shutter distortions parametrically as homography mixtures which are used to unwarp the distortions present in the original.

Our algorithm proceeds as shown in fig. 2. For a given video, we perform motion estimation, by first matching image corner features across frame pairs to obtain potential matches (section 3.1). After outlier rejection, we obtain a parametric model for motion and rolling shutter distortion between frames by fitting our homography mixtures to these matches (section 3.2). We also estimate a 4 degree of freedom similarity that is stabilized over time to account for global shake using the approach of Grundmann et al. [7],

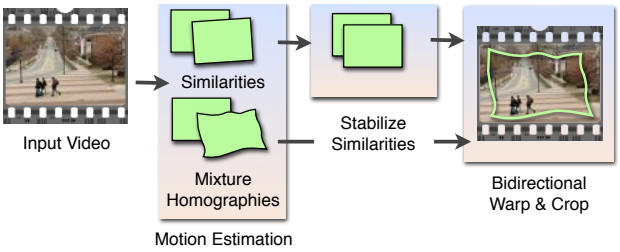


Figure 2: Overview of our algorithm.

resulting in per-frame crop window. Finally, the estimated homography mixtures are used to unwarp the rolling shutter distortions and the stabilizing crop is applied (section 3.4).

3.1. Feature extraction

To model the rolling shutter distortions between frames via parametric homography mixtures, we require matching image locations across frames. We follow the standard procedure by tracking KLT feature points using OpenCV to obtain sparse feature matches across frame pairs [2].

In contrast to image registration of undistorted data, we require dense coverage of high-quality features to model the wobble distortions across rows. To obtain dense coverage, we propose an adaptive version Shi and Tomasi's feature extraction [14]. The original algorithm determines corners at pixel locations where both eigenvalues of the 2nd moment matrix are above a pre-defined threshold. This threshold is usually chosen w.r.t. the maximum eigenvalue across all pixels, effectively imposing a frame-global threshold. We observed that this generally results in very few features within low textured regions such as sky or road because the foreground is highly textured, skewing the threshold unduly. We mitigate this issue by dividing the image into a grid of 4x4 equally sized bins, exercising a local threshold within each bin. To achieve scale independence we subdivide the grid iteratively across 3 pyramid levels. The effect of this technique can be seen in fig. 3.

In the presence of rolling shutter distortions, classical methods for outlier rejection such as imposing a fundamental matrix or global homography constraint are not applicable, as their assumption of a perspective transform between frames is violated. Similar to our adaptive feature extraction, we perform outlier rejection locally within equally sized bins across the image domain. Specifically, we robustly estimate the mean translation m_t for each bin using RANSAC and reject features that deviate from m_t by more than 2 pixels. We use an initial bin size of $1/4$ of the frame size that is uniformly downscaled by a factor of 2 across 3 pyramid levels. The final set of inliers is the union across pyramid levels.

Figure 3: Uniform (left) vs. our adaptive features (right). Using a local threshold w.r.t. the maximum eigenvalue of the 2nd moment matrix within each bin of a grid in the image domain enables us to track many more features in low contrast regions, such a grass or sky. This is crucial for modeling the rolling shutter distortion across frames. Also shown is the crop window used for stabilization, as described in section 3.4.

3.2. Homography mixtures

To motivate our homography mixtures, we briefly review the imaging process using fig. 4. After tracking and outlier rejection, for each frame pair (F_i, F_{i+1}) we have obtained a set of matching feature locations. For the subsequent discussion, we consider the matching feature pair (x, y) pictured in fig. 4. Both features are assumed to image the same 3D location X and are expressed in projective space $\mathbb{P}^2$.

In case of a global shutter, each row of a frame F_i is imaged at the same time T_i. Therefore, (x, y) are related by $x = P_i X$, $y = P_{i+1} X$, where P_i and P_{i+1} represent the corresponding projection matrices. Each projection matrix can be decomposed into an intrinsic camera matrix K_i and the camera center's origin t_i and orientation R_i at frame i, *i.e.* $P_i = K_i[R_i|t_i]$. In case of pure rotation ($t_i = t_{i+1} = 0$), the projection matrices are invertible and both frames are related by the relationship

$$x = P_i P_{i+1}^{-1} y = K_i R_i R_{i+1}^T K_{i+1}^{-1} y \quad \Rightarrow x = H_{i,i+1} y, \tag{1}$$

where $H_{i,i+1}$ is a 3x3 homography [9]. A similar linear relationship for x and y holds in case of non-zero translation if the scene is approximately in one plane or at infinity.

In case of rolling shutter, P_i and P_{i+1} are not frame-global but vary across rows. In this example, we try to recover the camera position at times $T(s_x)$ and $T(s_y)$ when image rows s_x and s_y of x and y were read out. Without loss of generality, we set $T_i = 0$, the read-out time of each row can be determined from its index:

$$T(s_x) = \frac{s_x(1-d)}{N} \in [0,1] \text{ and } T(s_y) = \frac{N + s_y(1-d)}{N},$$

where d is the camera dependent inter-frame delay, *i.e.* the time passing between the read-out of the last row N and the first of the next frame w.r.t. the frame period. Therefore, we adopt the simplified notation $P(s_x)$ and $P(s_y)$, to denote the camera position at times $T(s_x)$ and $T(s_y)$.

Current approaches to obtain $P(s_x)$ and $P(s_y)$ can be categorized into interpolation and regularization tech-

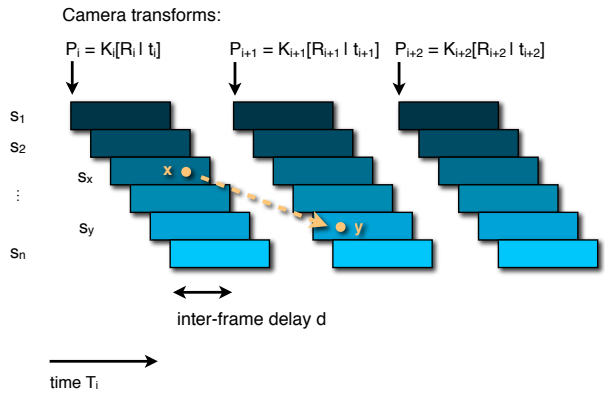


Figure 4: Motivation for homography mixtures. Matching feature location (x, y) imaging 3D location X, are related by $x = P_i X$, $y = P_{i+1} X$, in case of a global shutter. In case of rolling shutter, P_i and P_{i+1} vary across rows, depending on the corresponding scan lines s_x and s_y. Please see section 3.2 for details.

niques, each assuming piece-wise smoothness of the camera motion across rows.

Interpolation techniques: Liang et al. [11] use an interpolating translation model in the image domain ($K = I$), resulting in $P_i = [0 \mid t_i], P_{i+1} = [0 \mid t_{i+1}]$ which are globally estimated (translations are actually defined for the middle scanline, however we shift this to the first for ease of explanation.) The translation at row s_x is then given by $P(s_x) = [0 \mid q(T(s_x), t_i, t_{i+1})]$, where q is a Bezier curve interpolating between translations t_i and t_{i+1}. Forssen and Ringaby [5] extend this model to interpolate the rotation matrices instead, *i.e.* $P_i = K[R_i \mid 0], P_{i+1} = K[R_{i+1} \mid 0]$ with unknown rotations R_i and R_{i+1} and constant camera matrix K. Interpolation between rotation matrices is performed using spherical linear interpolation (slerp): $P(s_x) = K[\text{slerp}(T(s_x), R_i, R_{i+1}) \mid 0]$.

Regularization techniques: Baker et al. [1] uses a per row translation model in the image domain (K=I), independently estimating $P(s_j) = [0 \mid t_j]$ for each scanline s_j. L1 regularization is used to obtain piece-wise smoothness across rows, i.e. $|P(s_j) - P(s_{j-1})|$ is optimized to be small.

Homographies mixtures: Our homography mixtures can be regarded as generalization of above interpolation techniques to local homographies with additional regularization for improved stability. Note that, we can rewrite eq. (1) as $x = H_i H_{i+1}^{-1} y$, substituting $K_i R_i$ with an unknown homography H_i. In the case of rolling shutter the homographies depend on the row indices s_x and s_y resulting in the relation:

$$x = H(s_x) H(s_y)^{-1} y. \tag{2}$$

Note, this relation in not limited to the case of zero translation, but also holds if the scene is approximately in one plane or lies at infinity. We simplify eq. (2) by making the

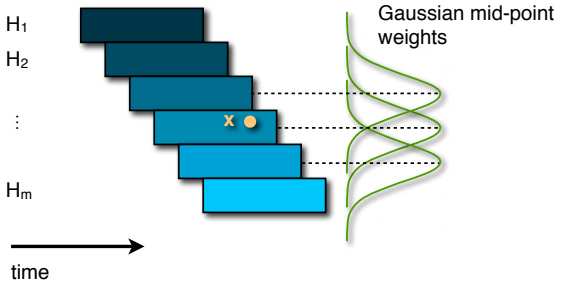


Figure 5: Homography mixtures defined over blocks of scanlines. To avoid discontinuities across scanlines the homography H_x for point x is given as mixture $H_x := \sum_{k=1}^{m} H_k w_k(x)$, with $w_k(x)$ being a gaussian weight centered around the middle of each scanline block k

assumption that all pixels within the vicinity of row s_x get mapped to row s_y, *i.e.* the relationship in eq. (2) only depends on the row index s_x. This assumption holds for arbitrary translations and small changes in scale, perspective and rotation, suited for the small inter-frame motion of the camera center in video. We therefore obtain:

$$x = H_x^{-1} y, \text{ with } H_x \sim H(s_x) H(s_y)^{-1}.$$

For efficiency and stability reasons, we estimate H_x for blocks of scanlines, as opposed to each scanline separately (estimation of homographies from collinear points is degenerated). Particularly, we partition the image domain in $m = 10$ blocks, resulting in 10 unknown homographies $H_k, k = 1..m$ needed to be estimated per frame to model the rolling shutter distortions. To avoid discontinuities across scanline blocks we smoothly interpolate the homographies using Gaussian weights as shown in fig. 5. The homography for point x is defined as mixture

$$H_x := \sum_{k=1}^{m} H_k w_k(x), \tag{3}$$

where $w_i(x)$ is a gaussian weight centered around the middle of each scanline block i. We use uniform sigma of 0.1 w.r.t. the frame height. Alternatively, to achieve interpolation behavior (gaussian weights only amount to approximation), one could use cubic hermite spline weights. We experimented with Catmull-Rom splines [3] and found them to be slightly less robust, when a scanline block contains very few features due to lack of texture. We believe this is caused by the fixed number of taps, as opposed to the exponential decaying gaussian weights, which extend across the whole frame. An illustrative example is shown for the translation component in fig. 8.

3.3. Estimation of mixtures

To fit a homography mixture H_k to a set of normalized matches $(x_i, y_i) \in [0, 1] \times [0, 1]$, we generalize the normal-

ized direct linear transform (DLT) [9] to mixture models. Specifically, for a match $(x, y) = ([x_1, x_2, 1]^T, [y_1, y_2, 1]^T)$ expressed as 3D vectors within the projective space $\mathbb{P}^2$, equality after transformation only holds up to scale, *i.e.*

$$0 = y \otimes H_x x = y \otimes \sum_{k=1}^{m} H_k w_k(x) x = \sum_{k=1}^{m} w_k(x) \cdot y \otimes H_k x, \tag{4}$$

where $\otimes$ denotes the cross product, and $w_k(x)$ is a known quantity, as it only depends on x and the fixed middle position of block k. Using the general DLT [9], we transform the expression $y \otimes H_k x$ to a set of 2 linear independent equations:

$$A_x^k h_k := \begin{pmatrix} 0^T & -x^T & y_2 x^T \\ x^T & 0^T & -y_1 x^T \end{pmatrix} h_k,$$

where h_k is the vector formed by concatenating the columns of H_k. We can then solve for eq. (4) by combining the above linearities for all mixture models k, yielding a $2 \times 9k$ linear constraint

$$\underbrace{\begin{pmatrix} w_1(x)A_x^1 & \dots & w_k(x)A_x^k \end{pmatrix}}_{:A_x} \underbrace{\begin{pmatrix} h_1 \\ \vdots \\ h_k \end{pmatrix}}_{:=h} = A_x h = 0. \tag{5}$$

Aggregating all linear constraints A_x for each feature match (x, y) yields an homogenous linear system, which can be solved for under the constraint $||h||_2 = 1$ using the SVD of A. Alternatively, the system can be transformed into a homogenous system by explicitly setting the bottom right element of each homography to 1, *i.e.* $h_k(3, 3) = 1 \; \forall k$, which is a reasonable choice for video, as the small inter-frame motions are virtually free of degenerated cases.

Robust estimation: While the choice of Gaussian weights $w_k(x)$ ensures smoothness across scanlines, we like to ensure that adjacent homographies h_k do not differ drastically. Furthermore, in case a block has fewer than 4 constraining matches, depending on the choice of the variance of the gaussian weights, eq. (5) can be under constrained and unstable to solve. We therefore propose to add a regularizer $\lambda||h_k - h_{k-1}||_2$ to the homogenous system, where we chose $\lambda = 1.5$.

To further improve robustness w.r.t. outliers, we iteratively solve for h using iterative least squares. After each iteration, we evaluate the geometric error $e_x := ||y \otimes H_x x||_2$, which is used to scale A_x in eq. (5) by the inverse error $\frac{1}{e_x + \epsilon}$. As residual wobble for high contrast regions is more noticable, we further chose to scale the inverse error by the color variance (expressed in Lab color space) of its surrounding patch, effectively approximating a patch-based registration error. An example is shown in fig. 6.

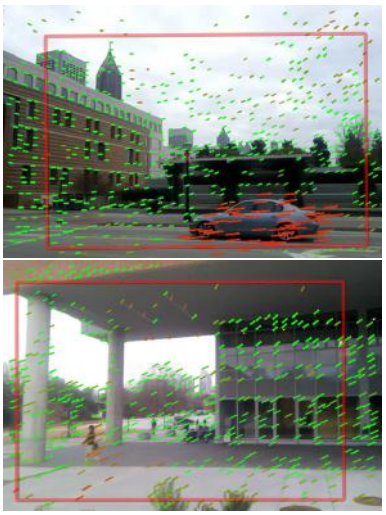


Figure 6: Outlier robust homography mixture estimation using IRLS weighting. Features with weight > 1 (residual distance less than 1 pixel) shown in green, features with weight $<< 1$ (residual distance considerably larger than 1 pixel) shown in red, using smooth interpolation in-between. Our technique successfully discounts foreground motion *e.g.* caused by moving objects or articulated bodies.

Reduced mixture models: One might ask, to which extent the different parameters (translation, affine and perspective) of a homography mixture vary across scanline blocks, *i.e.* what the effective minimum number of degrees of freedom is. To answer this question, we measured the variance of each homography mixture parameter across scanline blocks for two videos, normalized w.r.t. to its mean. The result is shown in fig. 7 for a parametrization of a general homography h as

$$h = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & 1 \end{pmatrix}. \tag{6}$$

It can be seen that perspective (h_7, h_8) and scale (h_1, h_5) can be regarded constant, while the parameters varying most across scanline blocks are translation (h_3, h_6) and skew (h_4).

Therefore, we propose two reduced mixture models of $6 + 2k$ and respectively $4 + 4k$ degrees of freedom:

$$H_k = \begin{pmatrix} A & t_k \\ w^T & 1 \end{pmatrix}, \text{and } \hat{H}_k = \begin{pmatrix} a & b_k & t_k^x \\ c_k & d & t_k^y \\ w_1 & w_2 & 1 \end{pmatrix}. \tag{7}$$

Here A is a frame-global 2x2 affine matrix, $w^T = (w_1, w_2)^T$ is the frame-constant perspective part and t_k is a block-varying translation. Likewise, a and d in $\hat{H}_k$ are frame-global scale parameters. These reduced models have the benefit of faster estimation and greater stability due to fewer degrees of freedom. We used the model $\hat{H}_k$ in all our

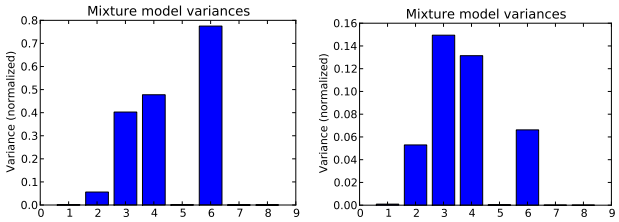


Figure 7: Normalized variance for each parameter of our homography mixtures across scanline blocks for two different videos. Shown are the 8 dof of a 3x3 homography h using the parametrization of eq. (6). Normalization is performed w.r.t. each parameter's mean. It can be seen that perspective (h_7, h_8) and scale (h_1, h_5) are nearly constant, while translation h_3, h_6 and skew h_4 have high variance.This motivates our reduced mixture model.

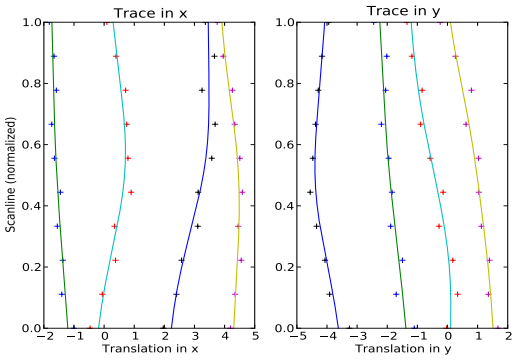


Figure 8: Example of block dependent translations t_k (see eq. (7)) shown as crosses and smooth trace obtained by interpolating the block-dependent translation via gaussian weights.

experiments, however H_k performs only marginally worse, and should be chosen if real-time performance is desired. An example plot of the block dependent translations t_k is shown in fig. 8.

3.4. Joint video rectification and stabilization

Using our computed homography mixtures we can perform rectification of the original video, effectively removing rolling shutter artifacts. To perform additional video stabilization, we implemented the video stabilization framework of Grundmann et al. [7] as it allows us to replace their frame registration method with our homography mixtures. As shown in fig. 2, for a given input video, for each frame-pair we estimate our homography mixtures H_n and additionally 4 degree of freedom similarities S_n (translation in x and y, scale and rotation). We stabilize the similarities using [7]. This results in a crop transform for each frame B_n indicated in red in fig. 6 and fig. 3.

To account for distortions beyond similarities, [7] proposed a bidirectional warping method. In particular, the computed crop transform B_n can be decomposed into $B_n = R_n S_n$, with S_n being the underlying similarity and R_n a residual. If perfect stabilization can be achieved, R_n is zero, *i.e.* the crop undoes the camera motion. However, due to the additional constraint that the crop rectangle has to stay within the frame, this is generally not the case. [7] proceeds by replacing S_n with a homography, instead we chose to replace S_n with our homography mixtures H_n, yielding a per-frame rectification and stabilization warp $\hat{B}_n = R_n H_n$. [7] address potential error accumulation over time using bidirectional warping of the frame by $\hat{B}_n$ w.r.t. equidistant spaced keyframes. We extend on their approach by using adaptively spaced key-frames to minimize potential distortion. In particular, for a frame interval $F_i, F_{i+1}, ..., F_k$, we compute the camera path w.r.t. origin F_i as homographies $H_1, H_2, ..H_k$. Our goal is to select H_l, $l = 1..k$ with the least non-rigid distortion as the next key-frame. To this end, each H_k is scored using 4 rigidity measures: Skew and change in aspect ratio (obtained by applying QR decomposition to H_k), modulus of perspective and average feature residual after registration. Considering the variance of each measure across frames, rigidity is defined using a normal distribution around mean zero (respectively mean one for aspect ratio). Lastly, assuming independence of the four measures, H_l is found at the frame $l = 1..k$ of highest probability, *i.e.* highest rigidity.

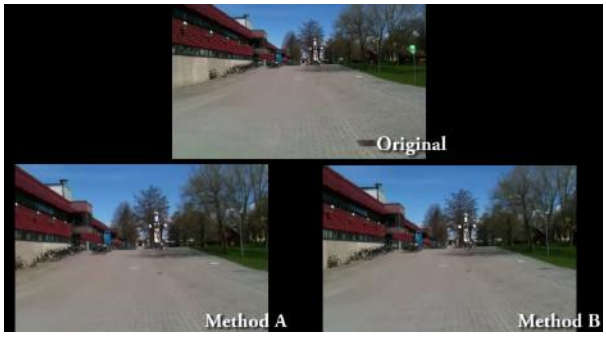
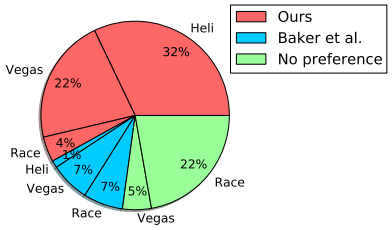


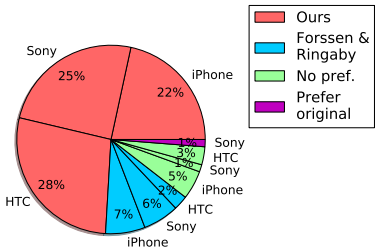
Figure 9: Layout of our user study. Users are presented with the original at the top and the results of two methods, labeled blindly as 'A' and 'B'. User is asked to chose among 4 choices: Prefer A, prefer B, no preference or prefer original.

4. Results

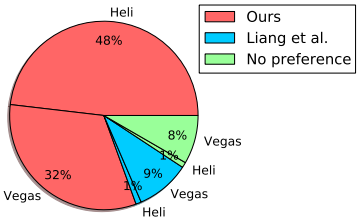
To evaluate our results qualitatively and compare to the results of six other authors, we conducted a user study with 54 participants. As shown in fig. 9, each participant is shown the original and two results after rolling shutter removal, labeled blindly as "Method A" and "Method B". Users were asked to choose which of the two presented methods reduces wobble and shake best. We asked users to disregard differences in aspect ratio, contrast or sharpness, as we compiled the videos from several authors and sources, each one using different video codecs and/or further post-processing which makes uniform treatment diffi-



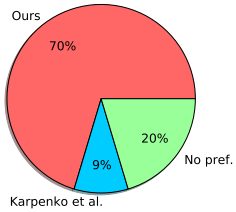
(a) Comparison to Baker et al. [1] on Helicopter, Vegas and Race sequence.



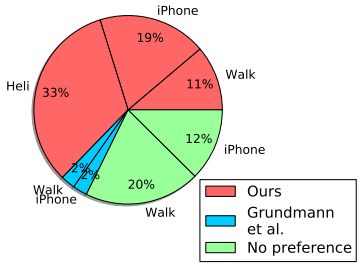
(b) Comparison to Forssen and Ringaby[6] on iPhone, Sony and HTC sequence.



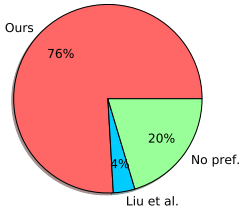
(c) Comparison to Liang et al. [11] on Helicopter and Vegas sequence.



(d) Comparison to Karpenko et al. [10] on Fountain sequence.



(e) Comparison to Grundmann et al. [7] on Walk, iPhone and Helicopter sequence.



(f) Comparison to Liu et al. [12] on Walking sequence.

Figure 10: Results of our user study consisting of 54 participants. We compare our algorithm to other authors on videos taken from their papers (see top row for thumbnails). Users are shown original and two results (ours vs. other author's, labeled *blindly* as method A and B). Users are asked which method they prefer (if any) w.r.t. reducing wobble. Charts indicate user choices averaged over all tested sequences (ranging from 1 to 3 videos, depending on other author's presented results). Also shown are individual results for sequences. Please see text for detailed discussion.

cult. In particular, users were presented with four choices for each video: (a) Prefer Method A, (b) Prefer Method B, (c) No preference - methods perform equally well and (d) Neither - prefer the original.

We compare our approach to six current state-of-the-art methods. Three of those methods are specifically designed to perform rolling shutter removal using visual features alone and require prior calibration as described in section 2: Baker et al. [1], Forssen and Ringaby [6], Liang et al. [11]. Further, two methods treat rolling shutter distortions as noise or as a global distortion: Liu et al. [12], Grundmann et al. [7]. We also include the approach of Karpenko et al. [10] which uses dedicated hardware in form of gyroscopes to supplement the visual estimation task.

For each other method, we selected a reasonable subset of rolling shutter distorted videos that were presented in that work. The thumbnails and labels for each video are shown at the top of fig. 10 and the aggregated responses of our user study are shown below. In general, the majority of all users showed strong preference towards our results when compared to other methods. This preference is even more pronounced when we only account for those users that actually showed a preference. We discuss the results w.r.t. each method in detail below.

Compared to Baker et al. [1], 58% of all users preferred our result, 15% preferred Baker et al. and the remaining ones indicated no preference. As shown in fig. 10a, the majority of no preference votes were cast for the "race" video. On the other two videos "helicopter" and "vegas", users prefered our solution by large margins. Note that Baker et al. 's approach requires the inter-frame delay to be known, where our approach does not require this information.

In fig. 10b, we compare to Forssen and Ringaby[6]. In general, 75% of all users prefer our method with less than 10 % showing no preference or preferring the original. The results are quite similar across the three tested

Figure 11: Scenarios for qualitative evaluation. We chose 4 different scenarios, shown from left to right: panning, walking forward, sidestepping and large depth variation. Each scene was recorded using 4 different cameras. Please see text and accompanying video.

videos. It should be noted that Forssen and Ringaby require a calibrated camera and a priori known inter-frame delay, whereas our approach does not require or use this information.

Compared to Liang et al. [11], who model rolling shutter as a global affine transform, 80% of all users preferred our results (fig. 10c). In comparison to Karpenko et al. [10] 70% preferred our result, and 20% indicated no preference (fig. 10d). Note, that Karpenko et al. determine the camera motion from gyroscopes instead of feature tracks.

The remaining two approaches we compared to are primarily video stabilization methods, that are somewhat robust to rolling shutter artifacts. Compared to Grundmann et al. [7], 63% preferred our results, while a considerable amount showed no preference (32%, fig. 10e). The results of [7] for the sequences "iPhone", "walk" and "helicopter" were obtained using the freely available online implementation on YouTube. Most votes indicating no preference were cast for the"iPhone" and "Walk" videos, both of which are mostly affected by frame-global skew. On the "helicopter" video however, which suffers mainly from wobble, all of the 54 users preferred our solution. Lastly, we compare to Liu et al. [12] in fig. 10f, where 76% prefer our result, while 20% show no preference.

In addition to the user study, we qualitatively evaluated the robustness and reproducibility of our method across different cameras. Specifically, we evaluated 4 cameras, among them 3 mobile phones without stabilization (iPod, Nexus 1 and Nexus S) and one mobile phone with gyro based stabilization (iPhone4S) across 4 different challenging scenarios, shown in fig. 11. Each scenario was recorded using each camera. Our method proved robust to significant foreground motion, changes in depth, high and low frequency bounces and wobble. We showcase the results in the accompanying video.

5. Summary

In this work, we presented a novel, calibration-free rolling shutter removal technique, based on a novel mixture model of homographies which faithfully models rolling shutter distortions. Our technique has the significant practical advantage that it adapts to the camera, rather than requiring a calibration procedure as previous approaches, resulting in a substantially increased range of applicability. In addition, our method is highly efficient (5 - 10 fps) while being robust to foreground motions and various challenging scenarios. We conducted a thorough evaluation using various cameras and settings as well as a user study, which showed that the majority of users prefer our results compared to other recent efforts. Our method can fail when a scene is composed of layers with significant differences in depth that cannot be adequately modeled by homographies or if the visual signal is too degraded (*e.g.* blur, missing features). In this case, supplementing the visual signal with gyroscope information should prove helpful.

References

[1] S. Baker, E. P. Bennett, S. B. Kang, and R. Szeliski. Removing rolling shutter wobble. In *IEEE CVPR*, 2010. 2, 4, 7

[2] G. Bradski and A. Kaehler. *Learning OpenCV*. O'Reilly Media Inc., 2008. 3

[3] E. Catmull and R. Rom. A class of local interpolating splines. *Computer aided geometric design*, 1974. 4

[4] W.-H. Cho and K.-S. Hong. Affine motion based cmos distortion analysis and cmos digital image stabilization. *IEEE Transactions on Consumer Electronics*, 2007. 2

[5] P.-E. Forssén and E. Ringaby. Rectifying rolling shutter video from hand-held devices. In *IEEE CVPR*, 2010. 1, 2, 4

[6] P.-E. Forssén and E. Ringaby. Efficient video rectification and stabilization of cell-phones. *Int. J. Comput. Vision*, June 2011. 2, 7

[7] M. Grundmann, V. Kwatra, and I. Essa. Auto-directed video stabilization with robust l1 optimal camera paths. In *IEEE CVPR*, 2011. 1, 2, 6, 7, 8

[8] G. Hanning, N. Forslöw, P.-E. Forssén, E. Ringaby, D. Törnqvist, and J. Callmer. Stabilizing cell phone video using inertial measurement sensors. In *IEEE International Workshop on Mobile Vision*, Barcelona, Spain, 2011. 1, 2

[9] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004. 3, 5

[10] A. Karpenko, D. Jacobs, J. Baek, and M. Levoy. Digital Video Stabilization and Rolling Shutter Correction using Gyroscopes. *Stanford CS Tech Report*, 2011. 2, 7, 8

[11] C.-K. Liang, L.-W. Chang, and H. H. Chen. Analysis and compensation of rolling shutter effect. *IEEE Transactions on Image Processing*, 2008. 2, 4, 7, 8

[12] F. Liu, M. Gleicher, J. Wang, H. Jin, and A. Agarwala. Subspace video stabilization. In *ACM ToG*, 2011. 2, 7, 8

[13] J. Nakamura. *Image Sensors and Signal Processing for Digital Still Cameras*. CRC Press, Inc., 2005. 1

[14] J. Shi and C. Tomasi. Good features to track. In *IEEE CVPR*, 1994. 3