March 20, 2019

Mean-rotation aware rolling shutter correction

Chia-Kai Liang

Fuhao Shi

Mean-rotation aware rolling shutter correction

ABSTRACT

A rolling shutter is a method of image capture that captures images scanline by scanline. Consecutive scanlines are captured at slightly different times. If fast relative movement exists between the camera and the world, e.g., during the capture of a fast-moving propeller, the captured scene is distorted. In particular, straight lines may look curved or tilted, since different segments of the straight line are captured at different moments of time. In some situations, a front-facing camera captures images of relatively static faces in the foreground of the image, even as the background experiences high velocity, e.g., due to camera panning. In this case, traditional rolling shutter correction with a virtual global shutter causes foreground distortion. This disclosure describes techniques to correct rolling-shutter distortions in the background of an image while preserving the relatively distortion-free image foreground.

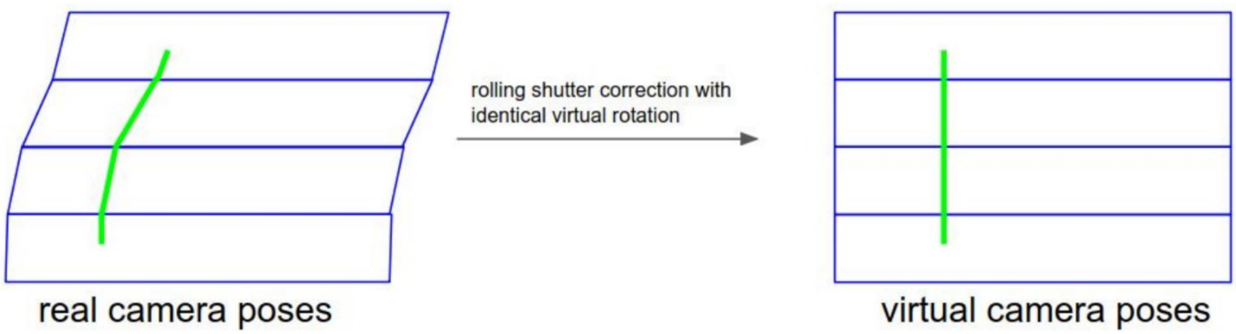
KEYWORDS

- Rolling shutter
- Scanline
- Gyroscope
- Camera pose
- Virtual scanline
- Image warping
- Motion model
- Major rotation axis
- Decomposed pose

BACKGROUND

A rolling shutter is a method of image capture that captures images scanline by scanline. Consecutive scanlines are captured at slightly different times. If fast relative movement exists between the camera and the world, e.g., due to camera panning, the captured scene is distorted. In particular, straight lines may look curved or tilted, since different segments of the straight line are captured at different moments of time.

Traditional rolling shutter correction methods divide the frame into groups of scanlines with approximately the same timestamp, calculate the real camera poses for each row, and warp each scanline to construct a single virtual camera pose. In this manner, scanlines in the corrected domain are virtually captured at the same timestamp and the rolling shutter distortion is corrected.

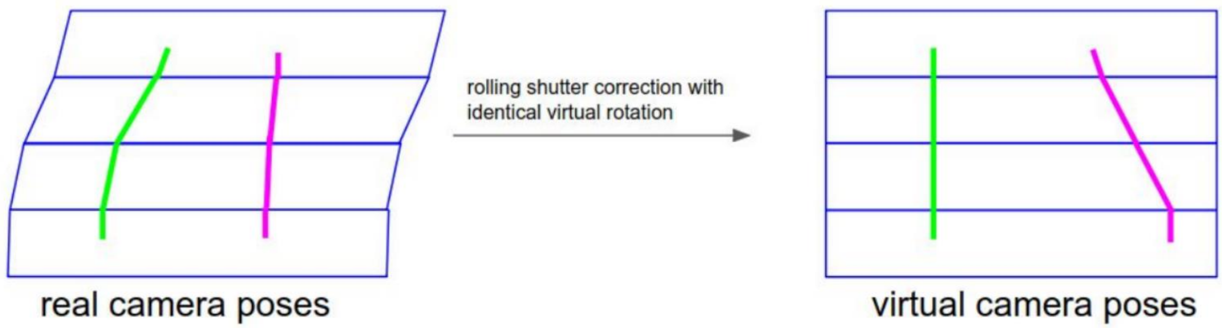


(A) (B)

Fig. 1: Traditional rolling shutter correction. (A) In the real camera pose, the green line, a straight object in the visual field, appears distorted. (B) After rolling-shutter correction, the green line appears straight.

This process is illustrated in Fig. 1, where real camera rows, each with different camera poses, are warped to a single virtual camera pose to correct rolling-shutter distortion.

However, traditional techniques for correcting rolling-shutter distortion fail under certain situations, e.g., when the background of the scene is moving at a relatively high speed with reference to the camera while the foreground is relatively stable. Such a situation can arise while taking front-facing videos of relatively static faces close to a panning camera. Such faces look normal and undistorted even under fast camera motion; however, in correcting background distortion, traditional rolling-shutter corrections introduce distortion to the faces, e.g., transform normal-looking faces to curved faces.



(A) **(B)**

Fig. 2: (A) A real-camera pose. The purple line is a static and undistorted foreground object such as a face. The green line is a fast-moving background object, distorted due to the rolling shutter effect. (B) After traditional distortion correction, the green background object is corrected for rolling-shutter distortion, but the purple foreground object is needlessly deformed.

This effect is illustrated in Fig. 2, where an originally undistorted foreground object (the purple line) is needlessly deformed in a bid to correct for distortions in background objects (the green line).

DESCRIPTION

This disclosure describes techniques to correct rolling-shutter distortions in the background of the image while preserving the relatively distortion-free foreground (comprising, e.g., faces) of the image.

Notation	Meaning
P_real[t,i]	Real camera pose for frame t at row i
P_real[t]	Real camera pose at the frame t's center
P_virtual[t,i]	Virtual camera pose for frame t at row i
P_virtual[t]	Virtual camera pose at frame t's center
R[P_from, P_to]	Rotation from pose P_from to P_to
Ax(R)	Rotation axis of a rotation R
Ag(R)	Rotation angle of a rotation R along its rotation axis
D[t]	Decomposed camera pose along the major motion direction from frame t at its center
S[t]	Smoothed camera pose for frame t at its center
Warp(P_from, P_to)	A homography transform that warps the frame from source camera pose P_from to target camera pose P_to
Slerp(P_from, P_to, r)	Spherical linear interpolation between poses P_from and P_to, with r∈[0,1] controlling the extent of interpolation between P_from and P_to

Table 1: Notation

Table 1 establishes certain notations that are used in this disclosure. In an example use of the notation of Table 1, traditional rolling shutter distortion correction may be described as:

```
Warp(P_real[t,i], P_virtual[t]), i = 1,...,N,
```

where N is the number of rows a frame is divided into. P_virtual[t] can be the real camera pose at the middle of the frame or an estimated camera pose that stabilizes the camera motion. In this context, a row of a frame represents a group of scanlines, e.g., one of a number of horizontal blocks that a video frame is divided into.

Per the techniques of this disclosure, a high-frequency gyroscope is used to correlate hand motion and the movement in the video. The rolling shutter distortion is corrected by replacing the single virtual camera pose with *a virtual camera pose at frame center with a consistent injected rotation between each virtual scanline*. In other words, the virtual camera poses for each row are different, and share the property that the rotation between each consecutive row pair is identical. In the notation of Table 1, the distortion correction is

`Warp(P_real[t,i], P_virtual[t,i]), i = 1,...,N,`

where `P_virtual[t,i]`, the virtual camera pose at row `i` for frame `t`, is given by

`P_virtual[t,i] = R_injected * P_virtual[t,i-1]`, and

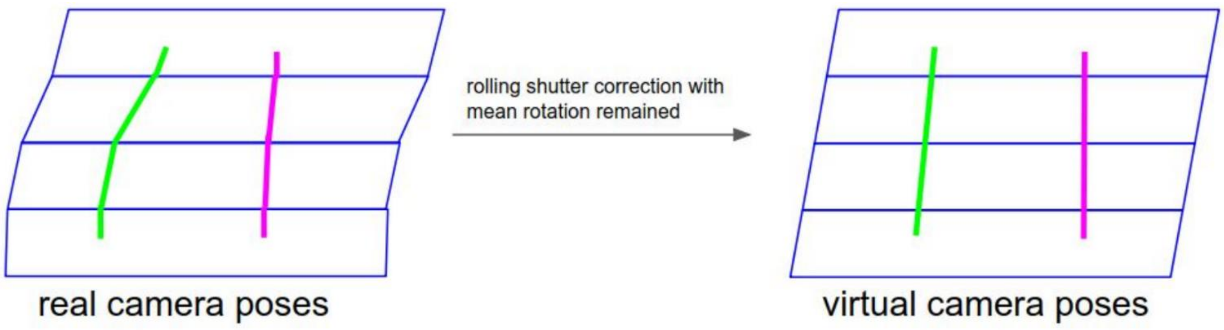
`P_virtual[t,i_center] = P_virtual[t]`.

Here, `R_injected` is the consistent rotation between consecutive rows, and `P_virtual[t, i_center]` is the pose at frame center, indicated by `i_center`, the index of the middle scanline. If the number of scanlines is even, `i_center` is set to the center of the middle two scanlines. For example, `i_center` equals 2.5 if the number of scanlines is 4. The value of `P_virtual[i_center]`, e.g., `P_virtual[t]`, can be the real camera pose at the middle of the frame or an estimated camera pose that stabilizes the camera motion.

`R_injected`, the consistent rotation between consecutive rows, has the following properties.

- It is a measure of the major rotation (usually user-intended), during a certain period.
- It changes smoothly across frames.
- It is close to the real camera rotations between each row pair.

Per the techniques of this disclosure, faces or other static foreground object are not distorted, e.g., sheared, shrunk, or enlarged, due to rolling-shutter correction, but random shakes are removed. However, some straight lines may look tilted from the real direction, which is a trade-off made to achieve faces that appear natural and undistorted.



(A) **(B)**

Fig. 3: (A) A real-camera pose. The purple line is a static and undistorted foreground object such as a face. The green line is a fast-moving background object, distorted due to the rolling shutter effect. (B) After mean-aware rolling-shutter distortion correction, the green background object is corrected for rolling-shutter distortion, while the purple foreground object continues to remain undistorted.

The results of applying the techniques are illustrated in the example of Fig. 3. A consistent rotation between each virtual scanline being injected into the virtual camera pose, the rolling shutter distortion in the background (the green line) gets corrected, while the foreground faces (purple line) remain undistorted.

In this disclosure, the words “pose” and “rotation” are used interchangeably, and they essentially mean a three-dimensional rotation. For example, a pose `P` can represent the integrated rotations from the beginning of the video (where the pose is initialized as identity), and a rotation `R` can represent the difference between two poses. Thus,

```
P[t] = R[t] * P[t-1] = R[t] * R[t-1]...* R[1],
```

where `R[1] = R_identity`.

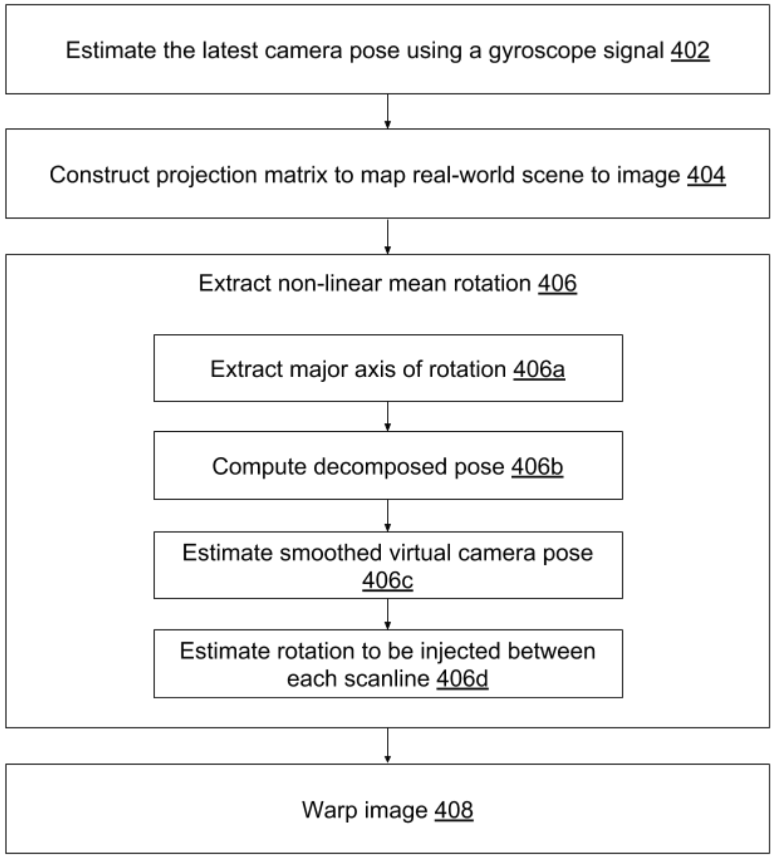


Fig. 4: Mean-rotation aware correction of rolling-shutter distortion

Fig. 4 illustrates mean-rotation aware correction of rolling-shutter distortion, per techniques of this disclosure. A gyroscope signal is used to estimate the latest camera pose (402), `P_real[t]`, at a relatively high frequency, e.g., 200 Hz. The camera pose can be represented, e.g., by a quaternion, or any other rotation representation. A gyroscope event handler can be used to fetch the gyroscope signal.

Given an input frame, its associated metadata (e.g., lens position and exposure time at each scanline), and the camera pose, a projection matrix is constructed (404) that maps the real world scene to the image `P_real[t,i]`. The process of constructing a projection matrix is known as motion-model construction.

The non-linear mean rotation is extracted (406) given the dense gyro and the smoothed camera pose history along the major motion directions. Extraction of the mean rotation may be done by a dedicated engine, whose output is the smoothed camera pose along the major direction of the current frame. Mean rotation extraction includes the following:

- *Extraction of major axis of rotation* (406a): The major rotation axis for the current frame with timestamp `t` is estimated from the gyroscope readout as follows. A set of camera poses `[P_real[1],..., P_real[N]]` within a time range `r`, say 1 second, and a step size 33 milliseconds (a frame length for 30fps video) is extracted. This time range can be `[t-r, t]` if pose extraction is done in real time and only gyroscope data up to time `t` is available. If pose extraction is delayed, it can be made non-causal (look-ahead), e.g., future data within the time range `[t-r/2, t+r/2]` can be included. The rotations between each pose pair are calculated as

  `R[P_real[t-1], P_real[t]] = P_real[t] * P_real[t-1]^-1.`

  The mean of the rotations is calculated as

  `Rm[t]=Mean(R[P_real[1],P_real[2]],...,R[P_real[N-1],P_real[N]]).`

  The major axis of rotation is extracted as `Ax(Rm[t])`.

- *Computation of decomposed pose* (406b): Given the major rotation axis `Ax(Rm[t])` and real rotation between current frame and previous frame

  `Rot = R[P_real[t-1], P_real[t]] = P_real[t] * P_real[t-1]^-1`,

  extract the rotation axis and angle as

  `Ax_real = Ax[Rot], Ag_real = Ag(Rot)`,

  and project the rotation axis along the major rotation axis as

```
R’[P_real[t-1], P_real[t]] = Ag_real*Dot(Ax_real,
Ax(Rm[t]))*Ax(Rm[t]).
```

The decomposed pose along the major direction D[t] can then be integrated as

```
D[t] = R’[P_real[t-1], P_real[t]] * D[t-1].
```

- *Estimation of smoothed virtual camera pose* (406c): A smoothed rotation `S[t]` for the current frame is estimated such that the rotation meets aforementioned properties, e.g., it follows the decomposed pose, it changes smoothly across frames, and it is similar to the real camera pose.

  The problem is formulated as an optimization of the following cost function:

  ```
  E = w_1 * E_follow_dcmp_pose + w_2 * E_follow_real_pose
                          + w_3 E_smoothness,
  ```

  where `w_1`, `w_2`, and `w_3` are weights that control how important each cost term is, and the three cost terms are:

  - `E_follow_dcmp_pose`, the decomposed pose-following term, which measures the difference between `S[t]` and `D[t]`. Intuitively, it nudges the virtual camera to rotate along the major direction.
  - `E_follow_real_pose`, the real camera pose-following term, which measures the difference between `S[t]` and the real camera pose `P_real[t]`. Intuitively, it prevents the virtual camera pose from being too different from the real camera pose.
  - `E_smoothness`, the rotation smoothness term, which measures the difference between the virtual camera poses of the current frame and the previous frames.

Other smoothness metrics, e.g., C0 smoothness (which stipulates that current and previous poses (S[t], S[t-1]) be similar) and C1 smoothness (which stipulates that the current and previous pose differences (S[t]*S[t-1]^-1, S[t-1]*S[t-2]^-1) be similar) can also be used. Rotation metrics such as the L2 difference between the quaternion representations (a 4D vector) can be used. These ensure that the virtual camera rotation changes smoothly. The cost function can be solved efficiently using numerical nonlinear solvers.

- *Estimation of rotation to be injected between each scanline* (406d): Given the optimized smooth virtual camera poses S[t], the injected rotation is estimated as a portion of the rotation between S[t-1] and S[t]. The portion ratio r depends on the length of the rolling shutter time, e.g., the scanning time from the top to the bottom of the frame, and the number of rows the frame is divided into. The calculation can be summarized as

  ```
  R_injected = Slerp(R_identity, R(S[t-1], S[t]), r),
  ```

  where R_identity is the identity pose to be interpolated from, R(S[t-1], S[t]) = S[t] * S[t-1]^-1 is the rotation between the current and previous virtual camera pose, and r = rolling_shutter_time/(N-1)/frame_length is the ratio of rolling shutter time between each row pair and the full frame length (33ms for 30fps video, for example). The Slerp() function is defined in Table 1.

An image warping engine warps the image (408) by loading the mapping output from the motion filtering engine, and mapping each pixel in the input frame to an output frame.

In this manner, the non-linear mean rotation of a sequence of video frames is extracted, and a consistent rotation calculated and injected into each virtual scanline. The rotation along the major motion direction remains unchanged after rolling-shutter correction, so that foreground objects such as faces are not distorted, while distortions caused by random shakes are removed. The techniques of this disclosure can be seamlessly combined with other video stabilization techniques.

CONCLUSION

A rolling shutter is a method of image capture that captures images scanline by scanline. Consecutive scanlines are captured at slightly different times. If fast relative movement exists between the camera and the world, e.g., during the capture of a fast-moving propeller, the captured scene is distorted. In particular, straight lines may look curved or tilted, since different segments of the straight line are captured at different moments of time. In some situations, a front-facing camera captures images of relatively static faces in the foreground of the image, even as the background experiences high velocity, e.g., due to camera panning. In this case, traditional rolling shutter correction with a virtual global shutter causes foreground distortion. This disclosure describes techniques to correct rolling-shutter distortions in the background of an image while preserving the relatively distortion-free image foreground.