A Systems Approach for Integrating Computer-Generated Imagery Into Rolling-Shutter Footage

R.L. Andersson
Andersson Technologies LLC

V2.1 Update April 13th, 2020

Introduction

Images from current CMOS cameras exhibit complex geometric distortions due to "rolling shutter" as an inherent result of the design of CMOS sensor chips.

While software can attempt to remove the effect of rolling shutter distortion from images, there are several unavoidable limitations:

- It introduces a resampling of the image, softening it.
- It introduces unavoidable image artifacts, especially around the edges of objects that are moving independently or at a greatly different distance from the background.

Given that rolling shutter distortion will be present, this white paper describes how to integrate computer generated imagery (CGI) with rolling shutter footage, based on existing workflow for lens distortion processing. This workflow requires appropriate software algorithms, which we will describe, for

- match-moving (available in SynthEyes 1208 and later) and
- rendering.

With this workflow, the source imagery, with rolling shutter distortion present, is match-moved to produce a match-move that corresponds to an idealized camera with no rolling shutter. New CGI elements are created using a renderer that simulates the rolling shutter of the original camera. These CGI elements can then be composited directly with the original unmodified source imagery. The final delivered footage is therefore as close as possible to the original.

Using the appropriate software, rolling shutter footage can be handled in existing lens distortion handling workflows with few changes.

A major point of this white paper is to advise renderer developers of the need to be able to simulate rolling shutter, and provide information on what is required.

Contents

Origin of Rolling Shutter

Rolling shutter is an imaging problem produced in popular "CMOS" cameras. This includes company-specific proprietary variations such as 3MOS, HyperMOS, etc, as well as other cameras such as RED that have other sensor names. It occurs just as much in expensive cameras as cheap ones. Only CCD cameras and cameras labeled as having a global shutter do not suffer from this problem.

CMOS cameras have the following essential elements:

1. a light-sensitive site for each pixel, where incoming photons produce electrical charges,
2. a capacitor that accumulates the electric charges for a shutter time
3. a switch that connects the capacitor to a shared readout bus whenever it is turned on,
4. amplifiers, analog to digital converters, and other processing circuitry that monitor the shared bus.

The exact details of #4 aren't relevant to the source of rolling shutter distortion. We've omitted other irrelevant details, such as the sensor's Bayer color filter pattern.

During operation, there are of course many sensor sites, organized into lines. All the lines constituting the image are read out, one after another, from top to bottom. As each line is read out, each pixel's capacitor is connected to a shared common bus so that it can be read and digitized. (Pixels could be read one at a time to a single bus, but a separate bus can handle each vertical column of pixels; there can be multiple amplifiers and A/D converters.)

The crucial part of this process is that reading the pixel onto the bus is a destructive process. After a pixel is read, the pixel is reset back to black, and it begins accumulating charge for the next video frame.

As you think about this, you should realize that there is no consistent shutter time period for the entire image. The top lines of the image are "taken" (integrated) over a physically different period of time than the bottom lines. The lines at the top are read and reset much earlier in time, the lines at the bottom read and reset much later.

Consider a 30 fps 1080p camera. The top line is read out then reset; it begins accumulating light and

charge for the next 1/30th of a second. As soon as the first line has been read out and reset, the second line is read out and reset, and it then begins accumulating light for 1/30th second. That continues for every line in the chip. Generally, for a 30 fps camera, it will take about 1/30th of a second to read out, one after another, all 1080 lines.

That means that by the time the bottom, 1080th, line is read out, almost a full 1/30th of a second has gone by. It will be accumulating light for a period of 1/30th of a second with virtually no overlap with the 1/30th of a second that the top line is integrated over, which has almost completed by the time the top line starts. The last line of the image has more in common with the first line of the *next* frame.

Consequences of Rolling Shutter

This wreaks havoc on the geometry of the image. Depending on how the camera is moving during the 1/30th of a second, and what it is looking at, an alarming number of geometric distortions will be introduced. Improperly shot footage with CMOS cameras will be objectionable even to lay human observers, because it does not correspond to our perception of the world---except in cartoons!

To the extent that the camera is panning left or right, the image will be skewed horizontally: a vertical pole will become slanted. To the extent that the camera is tilting vertically, images will be squashed or stretched vertically. If the camera is pushing in or pulling back, keystone distortions will result.

If the camera is vibrating, the image turns completely to jello. We rather famously lost an expensive helicopter shoot to this effect, see https://www.youtube.com/watch?v=GdjP3aumV1E

If there are cars driving by, one distortion can happen to the background, and a completely different distortion to the car! The same occurs for moving actors, with each actor's motion taken individually.

Common Misconceptions

There are some common misconceptions about rolling shutter distortion, and how to eliminate it.

- **It only happens if I *pan* too fast. Not true!** It happens whenever there is motion, whether it is the camera moving, or an object in the scene. And it happens no matter how the camera or object moves, for example vertical movements cause squash or stretch. All movement will cause some distortion; complex movements create complex distortion.
- **If I keep the camera movement slow, it won't happen at all. Not true!** It will be smaller, but it will still be there. You might not notice it as much, but if you are integrating CGI, we typically expect errors less than a pixel, which can be created by very small movement rates.
- **If I keep a fast shutter time, it won't happen. Not true!** The shutter rolls at exactly the same rate. Shutter time does not affect the amount of rolling shutter distortion. Your images will have less motion blur, and the crisper images may make the rolling shutter even easier to notice.
- **I can just skew the images a bit in 2D to fix them. Not true!** Depending on the camera motion, the distortion will take on different forms, and each independently moving object in the scene will be affected differently.

Altogether it seems a problem for a motion picture camera if it only works accurately for stationary

images.

Shooting Suggestions

For professional shooters, the usual tactic for CMOS is to make sure that the camera motion is slow in all directions, so that there is comparatively little motion from frame to frame (wasn't it supposed to be "moving pictures"?). CMOS cameras should be shot from heavy hard mounts such as dollies, cranes, etc. to minimize the chance of high-frequency vibrations.

Important: Use a CCD camera for aggressive hand-held shooting such as combat-style POV footage. See the warning and example about jello shots above if you doubt that.

Characterizing and Measuring a Camera's Rolling Shutter

The design of a camera affects the amount of rolling shutter distortion it produces, which can be characterized by a single number, its readout time, which is conveniently expressed as a fraction of the overall image frame time (the reciprocal of the frame rate). In most cases, CMOS cameras will exhibit near-worst-case performance, and the rolling-shutter value will be nearly one. When the manufacturer has taken steps to minimize the readout time, the rolling shutter value will be smaller, closer to zero, and the camera will be less sensitive to rolling shutter. CCD cameras have a rolling shutter value of exactly zero.

Fine point: to allow for camera footage with rolling shutter moving upwards, we allow the value to range from -1 to +1. Negative values can arise from stereo footage shot on mirror rigs, where one camera is mirrored vertically.

The rolling shutter value is specific to a model of camera. It will be exactly the same for all cameras of a specific model from a manufacturer, and possibly the same for multiple models from a given manufacturer. There is no need to determine the rolling shutter value for **each specific camera** (unlike lens distortion and decentering). It is likely, however, to vary for different frame rates or operating modes of a camera model.

The rolling shutter value can be determined by one of several methods:

- from the manufacturer's technical literature,
- from real footage with sufficient motion,
- by measurement from test footage, or
- from our website at https://www.ssontech.com/content/rollingValues.html if someone has previously measured it.

Data Sheet

Hopefully, as manufacturers become aware of this approach to working with rolling shutter, they will begin to supply the rolling shutter value in their data sheets, either directly or by specifying the image readout time, which can then be divided by the frame time.

For example, a manufacturer might specify an image readout time of 5 msec (0.005 sec) at 60 fps

(frames per second). At 60 fps, each frame is 1/60th second, or 16.667 msec. The rolling shutter value will be 5 / 16.667, ie 0.3.

Measurement (Live Shots)

SynthEyes 1806 and later can compute rolling shutter from actual (non-test) shots, when enabled from the Advanced Lens Controls panel. The accuracy of this method depends on how much the effect of rolling shutter can be distinguished (is different from) the effect of camera motion.

Measurement (Accurate)

The rolling shutter fraction can be quite accurately determined by shooting a scene with a rapid horizontal pan, thus exhibiting substantial rolling shutter, then having SynthEyes search to determine the rolling shutter value that minimizes the tripod-mode match-moving error. Typical suitable scenes are outdoors with much detail at various scales (but no blue sky!).

Be sure to have the camera leveled on a tripod, aimed straight across horizontally. Place the camera at the proper pivot point. Set the shutter time of the camera to be fairly short, 1/250th of a second or shorter, so that motion blur will not be a problem.

Computing the rolling shutter value was previously done by a script, but can now be done directly by the solver using the Advanced Lens controls. For the relevant tutorials and information to do this, see https://www.ssontech.com/tutes/tuterolling.html

If you use this technique to accurately measure rolling shutter on your camera, everyone would appreciate it if you can email us that information, so we can post it.

Measurement (Manual)

You can measure the rolling shutter value directly by shooting appropriate test footage, measuring successive frames using a tool such as Photoshop or SynthEyes, and dividing. This method is much less accurate and more difficult than the method above.

1. To shoot test footage, find a location with a large vertical black and white edge. The vertical edge must extend across the entire height of the camera image, from top to bottom, with plenty of margin, and should be as vertical as possible.
2. Put the camera leveled on a tripod, aimed straight across horizontally. Place the camera at the proper pivot point. Set the shutter time of the camera to be fairly short, 1/250th of a second or shorter, so that motion blur will not be a problem.
3. Shoot several takes that start with the camera looking at the vertical edge, then sweeping the camera back and forth rapidly so that the vertical edge sweeps rapidly across the image.
4. After shooting, transfer the footage to your computer, and open it in an application that allows you to measure the exact pixel coordinates of the cursor on any of the frames in the shot. In SynthEyes, you can read the cursor location on the status line at the bottom of the shot.
5. Start by checking the initial frames, where the camera is not yet moving. Measure the X coordinate of the vertical edge at top and bottom of the image. If the edge was perfectly

vertical, those X coordinates would be exactly the same, but in practice they won't be. Record the top-bottom difference (tilt) for later use.

6. Scrub forward into the footage to find a frame where the vertical edge sweeps at high speed across the center of the image. You'll notice that it is now slanted. Measure the X position of the vertical edge in the previous, center, and following frames to determine the average velocity per frame. You can perform this measurement at the top, vertical center, and bottom of the image and average the final results for more accuracy. The vertical edge should move at least 20 pixels per frame.
7. On that same (center) frame, measure the X position of the vertical edge at the top and bottom of the image. Subtract the pre-existing tilt, measured previously, from the top-bottom difference measured now, to obtain a slant value.
8. Compute the rolling shutter fraction by dividing the slant value by the frame-frame velocity.

You can understand this final calculation with an example. Suppose the edge is moving 50 pixels per frame, and the edge is slanting 45 pixels per frame. That means it is taking 45/50 ie 0.9 (90%) of the frame time to read out the image. If the edge is slanted only 5 pixels, the rolling shutter value would be 5/50 ie 0.1.

Approaches to CGI Integration with Rolling Shutter

Given that CMOS cameras are in wide use, there are several potential methods to be able to integrate visual effects with that footage:

a. try to shoot to keep the rolling-shutter low, and try to cheat the tracks and inserts to allow for the inevitable errors,
b. use a third-party software tool to try to correct the footage, or
c. compensate for the rolling shutter in the solve, producing a solve ¨for an 'ideal' camera.

The first choice can work out for small amounts of distortion and short shots to be tracked. The solver will adjust the solve to best match the distorted data, which winds up allowing inserts to appear at a correspondingly distorted location. Long shots, shots that loop back on themselves, and other shots with high self-consistency requirements will be substantial problems.

The second choice, a third-party tool, can be OK for simple images too. But, keep in mind that rolling-shutter causes a **permanent** loss of data in the image (this is unlike a lens distortion, which can be unwarped back into place). Any repair can be only a best guess at the missing data, and for that reason you will commonly see artifacts around edges, and residual distortion.

The third choice, producing a solve for an ideal camera, is the approach we are advocating in this white paper, and make available in SynthEyes.

Match-moving Rolling Shutter Footage

To compensate for the effect of rolling shutter, SynthEyes corrects the 2-D position of each tracker on every frame, adjusting its position to its (predicted) location at a single *reference time* that is common to the entire frame and all the trackers on it. By bringing all the tracker positions into alignment at a single

reference time, SynthEyes produces tracking data equivalent to that of a non-rolling-shutter (ie CCD) camera, had it been used instead.

There are a few key points to notice about this (correcting the tracker locations):

- there is never any “corrected” or “fixed” image required or produced, only the tracker data is changed;
- because each tracker is adjusted independently, different portions of the image can act completely independently, ie it is possible to correctly process images with multiple independently-moving objects;
- there is no limit on “how much” rolling shutter can be handled, ie the image can be slewing so rapidly that there would be large regions that would be completely un-imaged, if we were trying to “fix” the image, as long as the trackers were tracked properly (ie supervised as needed).

SynthEyes brings the tracking data into alignment at the reference time using simple linear interpolation. This is a reasonable approach:

- normally the tracking data changes relatively smoothly from frame to frame,
- random tracking data jitter will produce high-frequency error terms that will create incorrect intermediate locations if higher-order interpolation is used,
- in the event that tracking data is changing at such a rate that a non-linear interpolation would normally be indicated, that means it contains high-frequency components likely to create a jello shot.

If, in the future, some more advanced interpolation appears useful, it is easily added.

As a reference time, we use the time that corresponds to the middle line of the image during the rolling-shutter sweep, ie on a 1080 line image, the reference time corresponds to line 540, midway through the sweep. Each tracker’s position on every frame is adjusted to that same time, that of the center line. Using the center line minimizes errors during prediction: we never have to adjust a tracker position’s time by more than half a frame.

After modifying the tracker data, it all corresponds to that same reference time, and when the scene is solved for 3D, all the camera positions also correspond to that same time on each frame. The rolling shutter camera has been turned into an idealized CCD camera, at least for tracking.

In SynthEyes, you can turn on rolling shutter compensation and supply the rolling shutter value on the shot setup panel, either when the shot is opened or subsequently with Shot/Edit Shot. If you solve a shot initially without rolling shutter enabled, then enable rolling shutter and re-solve, you’ll typically see a substantial reduction in the overall RMS scene error.

When you solve a shot with one or more moving objects, and possibly a moving camera, the match-move comes out correctly, independent of the different rolling shutter effects affecting the different objects, because each tracker has been corrected independently.

Rendering with Rolling Shutter

In order to composite CGI generated for the ideal camera with the original rolling-shutter-based footage, you must render the CGI footage with simulated rolling shutter. At present, the capability to simulate rolling shutter is not widely available (see following section). This section is primarily intended for renderer developers, to rectify that.

Simulating rolling shutter is essentially a modified form of motion blur, where image lines at the top of the image are averaged over an earlier period of time than lines at the bottom of the image. The relative integration period within the frame can easily be calculated.

Using a frame-rate-invariant approach, we measure time in frames, ie frame 0, frame 1, frame 2, etc. Motion blur is normally calculated over a window determined by the shutter angle, ie for a 270 degree shutter the motion blur is integrated over a period of 0.75 frames.

To simulate rolling shutter, we need to pin this down more exactly. Frame *f* should be integrated over the period of time from *f - 0.375* to *f + 0.375*, ie over the same 0.75 frames but centered on the frame. (Adopting a centered convention is desirable when rolling shutter is included, as we'll see.)

Earlier, we defined the nominal reference time for a frame to be its center in the vertical direction, ie if the frame has 1080 lines, line 540 is the center and corresponds **exactly** to the given frame time *f*. (Some cameras may have rolling shutter within a line, and we might more precisely put the center at line 539.5, but these effects are negligible.)

For a rolling shutter value *rsv* and shutter angle of 270 degrees, the integration period for line 0 will extend from *f - 0.375 - 0.5*rsv* to *f + 0.375 - 0.5*rsv*. At line 540, the integration period is the original *f - 0.375* to *f + 0.375*. At line 1079, the integration period is *f - 0.375 + 0.5*rsv* to *f + 0.375 + 0.5*rsv*.

If we have a line *y, with y* ranging from 0 at top to *vres-1* at bottom, it should be integrated over the time period *f - 0.5*(shutter/360) + (y/vres - 0.5)* to *f + 0.5*(shutter/360) + (y/vres - 0.5)*. Since the rolling shutter value can range up to one and the shutter angle up to 360 degrees, notice that integration can be based on information all the way from the previous frame *(f-1)* to the following one *(f+1)*, but not beyond (in either direction). Negative rolling shutter values for mirrored cameras do not affect the required range. The centered approach we have taken in defining our parameters and intervals makes that possible, and in addition it minimizes undesirable extrapolation.

Software Packages Supporting this Approach

We are aware of the following software packages supporting this rolling shutter approach:

- SynthEyes, version 1208 or later
- Lightwave 11.5 or later
- Renderman, via a rolling-shutter simulation plugin

Though the Renderman plugin is available, we haven't seen it to verify what kind of timing parameters it provides, some some translation of parameters might be required.

For now, dedicated end users might simulate rolling shutter by rendering at a multiple of the frame rate,

then combining the subframes in varying amounts depending on the vertical position in the image.

Caveats and Warnings

To reiterate: the approach described here allows users to accurately integrate CGI into as-shot rolling-shutter images, even with multiple complex motions. **It does not and will not "fix" images exhibiting visible rolling-shutter or jello-shot artifacts**, which is impossible to do exactly, even in theory.

While shooting with CMOS cameras, extra attention should be paid to sources of vibration, and a camera mount used that eliminates vibration over 10-15 Hz. If a CMOS camera is subject to vibration over the Nyquist rate, ie ½ the frame rate, chaotic jello-shot images will result, and there is little prospect for obtaining a watchable result.

You can use the tabulated versions from our website or your own measurements to provide insight into what camera to select, and potentially what modes to use to minimize rolling shutter.

Conclusion

In this paper, instead of trying to fight a futile battle to eliminate rolling shutter artifacts, we've shown how to work with rolling shutter images to deliver the best possible results. In the long term, this should be a more productive approach. With appropriate match-moving and rendering tools, only some very small additional setup time is required.