Synchronization and Rolling Shutter Compensation for Consumer Video Camera Arrays

Derek Bradley Bradley Atcheson Ivo Ihrke Wolfgang Heidrich

University of British Columbia

Abstract

Two major obstacles to the use of consumer camcorders in computer vision applications are the lack of synchronization hardware, and the use of a "rolling" shutter, which introduces a temporal shear in the video volume.

We present two simple approaches for solving both the rolling shutter shear and the synchronization problem at the same time. The first approach is based on strobe illumination, while the second employs a subframe warp along optical flow vectors.

In our experiments we have used the proposed methods to effectively remove temporal shear, and synchronize up to 16 consumer-grade camcorders in multiple geometric configurations.

1. Introduction

Consumer camcorders are evolving as promising alternatives to scientific cameras in many computer vision applications. They offer high resolution and guaranteed high frame rates at a significantly reduced cost. Also, integrated hard drives or other storage media eliminate the need to transfer video sequences in real-time to a computer, making multi-camera setups more portable.

However, there are also a number of challenges that currently limit the use of such camcorders, especially in multi-camera and camera array applications. First, consumer camcorders typically do not have support for hardware synchronization. Second, most consumer cameras employ a "rolling" shutter, in which the individual scanlines use a slightly different temporal offset for the exposure interval (see, e.g. [22]). The resulting frames represent a sheared slice of the spatio-temporal video volume that cannot be used directly for many computer vision applications.

In this paper we discuss two different approaches for solving both the synchronization and the rolling shutter problem at the same time. The first method performs optical synchronization by using strobe illumination. Strobe lights create simultaneous exposure images for all cameras that can be used for synchronization. The simultaneous strobe flash also removes the rolling shutter problem, although the scanlines for a single flash are usually distributed across two frames (or fields, with interlacing).

Our second approach works in situations such as outdoor scenes, where strobe illumination is impractical. This method eliminates the rolling shutter shear by applying a warp along optical flow vectors to generate instantaneous images for a given subframe position. If the subframe alignment between multiple cameras can be determined using a synchronization event, this approach can also be used to synchronize camera arrays.

In the following, we first review relevant work on camera synchronization (Section 2), before we elaborate on the rolling shutter camera model on which we base our experiments (Section 3). We then discuss the details of our two synchronization methods in Section 4 and 5. Finally, we present results from our experiments in Section 6.

2. Related Work

Due to the time-shifted exposures of different scan-lines, rolling shutter cameras are not commonly used in computer vision. However, over the past several years, analysis of this sensor type has increased and a few applications have been described in the literature.

Rolling Shutter Cameras in Computer Vision. Wilburn et al. [22] use an array of rolling shutter cameras to record high-speed video. The camera array is closely spaced and groups of cameras are hardware triggered at staggered time intervals to record high-speed video footage. Geometric distortions due to different view points of the cameras are removed by warping the acquired images. To compensate for rolling shutter distortions, the authors sort scanlines from different cameras into a virtual view that is distortion free. Ait-Aider et al. [1] recover object kinematics from a single rolling shutter image using knowledge of straight lines that are imaged as curves.

Wang and Yang [20] consider dynamic light field rendering from unsynchronized camera footage. They assume that images are tagged with time stamps and use the known time offsets to first compute a virtual common time frame for all

978-1-4244-3993-5/09/$25.00 ©2009 IEEE

cameras and afterwards perform spatial warping to generate novel views. Camera images are assumed to be taken with a global shutter.

Rolling Shutter Camera Models and Image Undistortion. Although there are hardware solutions for the CMOS rolling shutter problem, e.g. [21], these are often not desirable since the transistor count on the chip increases significantly, which reduces the pixel fill-factor of the chip. Lately, camera models for rolling shutter cameras have been proposed, taking camera motion and scene geometry into account. Meingast et al. [14] develop an analytic rolling shutter projection model and analyze the behavior of rolling shutter cameras under specific camera or object motions. Alternatively, rolling shutter images can be undistorted in software. Liang et al. [11, 12] describe motion estimation based on coarse block matching. They then smooth the results by fitting Bézier curves to the motion data. The motion vector field is used for image compensation, similar to our approach described in Section 5, however we perform dense optical flow and extend the technique to a multi-camera setup to solve the synchronization problem as well. Nicklin et al. [15] describe rolling shutter compensation in a robotic application. They simplify the problem by assuming that no motion parallax is present.

Synchronization of Multiple Video Sequences. Computer vision research has been concerned with the use of unsynchronized camera arrays for purposes such as geometry reconstruction. For this it is necessary to virtually synchronize the camera footage of two or more unsynchronized cameras. All work in this area has so far assumed the use of global shutter cameras. The problem of synchronizing two video sequences was first introduced by Stein [18]. Since Stein's seminal work, several authors have investigated this problem. Most synchronization algorithms are based on some form of feature tracking [6]. Often, feature point trajectories are used in conjuction with geometric constraints relating the cameras like homographies [7, 18], the fundamental matrix [5, 17] or the tri-focal tensor [10]. The algorithms differ in how the feature information is matched and whether frame or sub-frame accuracy can be achieved. Most authors consider the two-sequence problem, but N-sequence synchronization has also been considered [5, 10].

A different approach to N-sequence synchronization has been proposed by Shrestha et al. [16]. The authors investigate the problem of synchronizing video sequences from different consumer camcorders recording a common indoor event. By assuming that in addition to the video cameras, the event is being captured by visitors using still cameras with flashes, they propose to analyze flash patterns in the different video streams. By matching binary flash patterns throughout the video sequences, frame-level synchronization can be achieved.

Stroboscopic Illumination. Stroboscopic illumination has been used to capture multi-exposure images. Classic examples include early photographic work by Harold E. Edgerton and Gjon Mili to capture high-speed events on film. Lately, computer vision techniques have used this principle to recover trajectories of high speed motions, e.g. Theobalt et al. [19] track the hand motion and ball trajectory of a baseball player. Linz et al. [13] recover flow fields from multi-exposure images to generate intermediate single exposure views and synthetic motion blur.

Summarizing, related work has been concerned with several aspects of the methods we are proposing in this paper. However, we demonstrate the novel use of these techniques for the realization of low-cost camera arrays with good synchronization characteristics.

3. Camera Model

Both of our synchronization methods target inexpensive consumer-grade video cameras and camcorders. In this camera segment, there has been a recent push to replace CCD chips with CMOS sensors. These sensors have a number of advantages, but can also introduce rolling shutter distortions that we aim to model and eliminate.

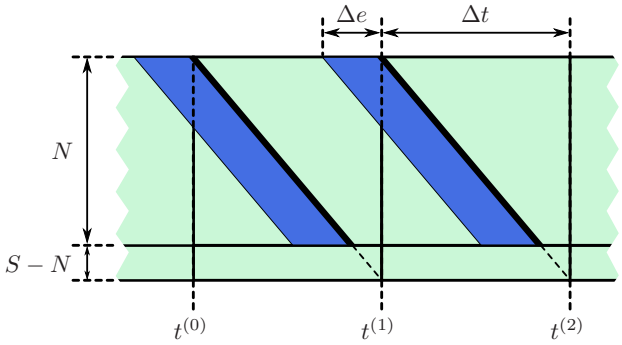

Figure 1. Rolling shutter camera model. Just-in-time exposure and readout of the individual scanlines creates a shear of the exposure intervals along the time axis. The slope of this shear is a function of the camera frame rate and the period is determined by the number of scanlines in the video format.

Rolling Shutter. To minimize buffer memory, the less expensive CMOS cameras read out each individual scanline from the sensor chip just in time for generating the video signal for that scanline. The exposure interval for each scanline starts some fixed time before that readout time, so that effectively the exposure interval is temporally offset for each individual scanline. More specifically, we can model the readout time $r_y^{(j)}$ for scanline y in frame j as follows (also see Figure 1):

$$r_y^{(j)} = t^{(j)} + \frac{y}{S} \cdot \Delta t = t^{(0)} + \left(j + \frac{y}{S}\right) \cdot \Delta t, \qquad (1)$$

where Δt is the frame duration (one over the frame rate), S the total number of scanlines per frame, and $t^{(j)}$ the readout time of the topmost (visible) scanline in frame j. Since CMOS sensors operate similar to RAM, we can model pixel readout as instantaneous for our purposes. The exposure interval for scanline y in frame j is then given as

$$E_e^{(j)} = \left[r_y^{(j)} - \Delta e,\ r_y^{(j)} \right], \tag{2}$$

where Δe is the duration of exposure (exposure time).

Note that the total number S of scanlines may be larger than the number N of *visible* scanlines. For example, the specification for high definition video [8] calls for $S = 1125$ total lines for video standards with $N = 1080$ visible lines. The extra 45 invisible lines correspond to the vertical synchronization signal. For standard definition video, the number of invisible scanlines is 39 ($S = 525$) in NTSC [9].

Interlacing. Most consumer cameras trade spatial for temporal resolution by recording the even and the odd scanlines in separate fields (interlacing).

The theoretical analysis from above can easily be adapted to interlaced video, where the temporal separation between fields is now $\Delta t/2$, and the number of visible and total scanlines is $N/2$ and $S/2$, respectively. Note that S is an *odd* number for most video standards, which means that the duration of a field is slightly different for odd and even frames. However, this difference is only a fraction of a percent of the field time, and will be ignored for our purposes.

Synchronization. For synchronization of multiple cameras, we assume that all cameras follow the same video standard, i.e. that Δt, S, and N are identical for all cameras, and that either all or none of the cameras use interlacing. These assumptions are easily met if all cameras in the array are the same model. A possible concern is the potential for slight differences in the frame rate across individual cameras. However, even inexpensive cameras appear to have very good accuracy and stability with respect to frame rate. In our experiments with up to 16 cameras and several minutes of video duration, per-camera differences did not appear to have a visible impact, see also Section 6.1 for a detailed experiment description.

4. Method 1: Stroboscope Illumination

Stroboscopes have long been used for obtaining instantaneous exposures of moving objects using standard cameras without rolling shutters (e.g. [19]). An extension of this approach to multi-camera systems results in multiple video streams that are *optically* synchronized through illumination. Unfortunately, this straightforward approach fails for rolling shutter cameras, which we address in the following.

With our stroboscope approach, we can solve the rolling shutter problem for individual cameras, or simultaneously solve the synchronization and rolling shutter problems for an array of cameras. With no ambient illumination, stroboscopes create simultaneous exposures for all cameras. However, with rolling shutters, the exposed scanlines are usually divided between two adjacent frames. In our technique, we combine two partially exposed frames to form a single synchronized exposure image for each camera. Since all scanlines are exposed by the flash at the same time, this method avoids the temporal shear usually caused by rolling shutter cameras.

Basic Approach. In the single camera setting, the camera starts recording a dark scene in a normal way. Stroboscopic illumination is then activated, creating the exposures. For now, we will assume the flash is instantaneous. The flash is captured by all scanlines that are exposing at the time of the flash. The number of scanlines that record the event is determined by the exposure time of the camera, Δe (recall Figure 1). In the most basic approach, we ensure that all scanlines record the flash by maximizing Δe (see Figure 2) creating a continuous exposure with respect to the camera. Due to the overlapping exposure windows of the scanlines in rolling shutter cameras, the strobe flash is usually split between two consecutive frames. The two frames containing the instantaneous exposure can be combined with addition, or in this simple case a maximum operator as we illustrate in Figure 2. Note that we use a binary marker grid simply as a basic scene for demonstrating the technique.

In a multi-camera setting, each camera independently captures the scene with the strobe illumination. The per-camera rolling shutter compensation as described above automatically synchronizes the array.

Virtual Frame Rate. Although the cameras record frames at a certain frame rate ($1/\Delta t$), the frequency of the strobe lighting creates a *virtual frame rate* for the video sequence. This is because one output frame is generated for each flash of the stroboscope. The maximum frequency that avoids double-exposure of scanlines is $1/\Delta t$. However, flashing at this frequency tightly packs the instantaneous exposures with no gap of nearly-black pixels between them. Leaving a gap between the exposures helps to separate them, especially in the case of minor drift if the stroboscope frequency cannot be set precisely to $1/\Delta t$. The simplest approach is to set the strobe frequency to half the camera frame rate ($1/2\Delta t$), creating a full frame of unexposed scanlines between every exposure. Note that the unexposed scanlines are also split between two consecutive frames, exactly like the exposed scanlines. If this reduction in temporal resolution is acceptable, then every pair of adjacent frames can be combined in the straightforward manner described above. If a higher virtual frame rate is desired, the frames can still be combined automatically with a little more computational effort to explicitly search for the unex-

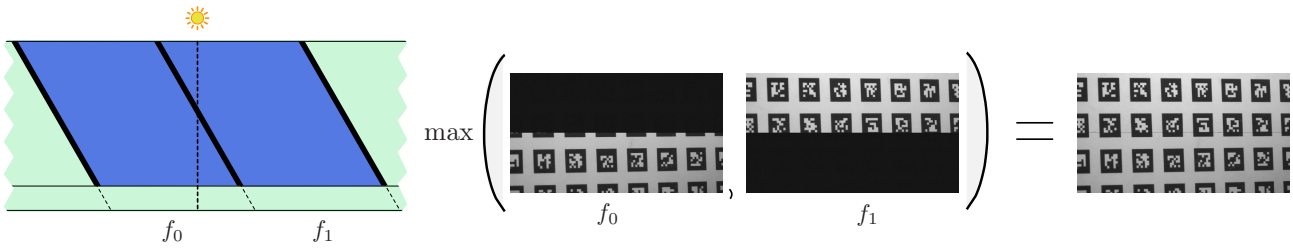

Figure 2. In rolling shutter cameras, consecutive frames that contain the instantly exposed scanlines are combined to make the final image.

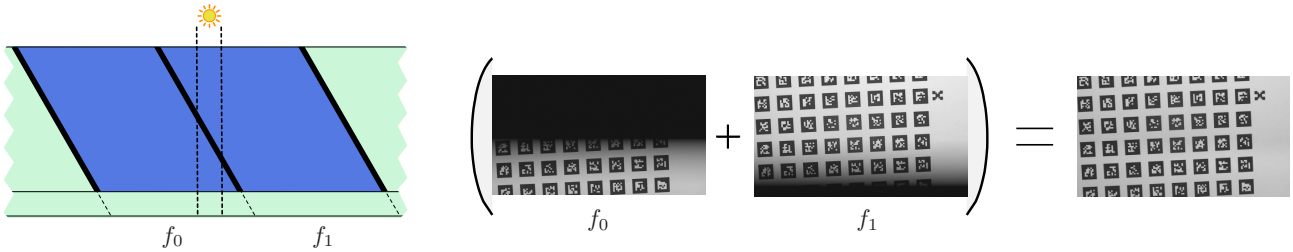

Figure 3. Increasing the flash duration creates a virtual exposure time. The exposed scanlines overlap with a ramp up at the begining and ramp down at the end. Summing the frames in linear space creates the final image.

posed scanlines that separate the frames. This technique is robust to any minor drift that may occur over time, if the strobe frequency cannot be set with high precision.

This method removes motion blur from the video frames, which allows us to capture very fast moving objects (see Figure 5). The downside of this approach is that an instantaneous strobe flash may not produce very much illumination, resulting in increased camera noise. We now describe how a controllable, longer flash duration allows us to trade off noise for motion blur.

Virtual Exposure Time. As we mentioned previously, varying the flash rate creates a virtual frame rate for the output sequence. We can also create a *virtual exposure time*, Δe_v, by varying the duration of the flashes. As we see in Figure 3, increasing the flash duration creates an overlap of exposed scanlines in consecutive frames. The amount of overlap is directly related to the virtual exposure time, i.e. the number of scanlines that can be read out during the flash duration. The exposure also ramps up at the beginning and ramps down at the end, though summing the two frames in linear space gives the correct final image. In practice the camera data is not linear, however many cameras follow a standard gamma curve which can be inverted before adding the frames. It can also happen that the exposed scanlines span three frames due to the scanline overlap, but the linear summation is still the same in this case.

Having a longer virtual exposure time creates less noisy images as the camera gain can be minimized. However, this comes at the expense of motion blur for fast moving objects. By allowing a controllable flash duration, the trade-off between noise and motion blur can be chosen depending on the application (see Section 6.3 for a detailed experiment).

Non-Continuous Exposure. In the most general camera model, the exposure time Δe, can also vary such that the camera is not recording continuously. In this case, summing the adjacent frames no longer gives a full-intensity result for all scanlines, since there is less overlap between the frames (see Figure 4). However, as long as $\Delta e_v > \Delta t - \Delta e$, we can artificially boost the gain per scanline to recover the frame. The per-scanline scale factor can be computed by recording a diffuse white surface. If the above inequality does not hold, i.e. either the camera exposure or the virtual exposure is too short, then there will be missing scanlines that cannot be recovered.

Interlaced Video. Thus far, our method has been described for non-interlaced video cameras, however it easily extends to interlaced video as well. Interlaced cameras separately record odd and even fields at twice the frame rate. This implies that a stroboscopic exposure is split between two fields rather than two frames. The two partially exposed fields can be combined into a single *half-frame* with the same technique described above. Note that we refer to this as a half-frame rather than a field since some of the scanlines are even and some are odd. The half-frame can then be converted to a full frame by interpolating the missing in-between scanlines. Although we only capture half the spatial resolution with each flash of the strobe, we gain up to twice the temporal resolution since the field-rate is twice the frame rate. This means that we can set the desired virtual frame rate and virtual exposure time, and then combine fields instead of frames at the cost of spatial resolution.

Our stroboscope illumination technique works effectively by starting the cameras first and the strobe lighting second. The first exposure flash can then be located in each camera,

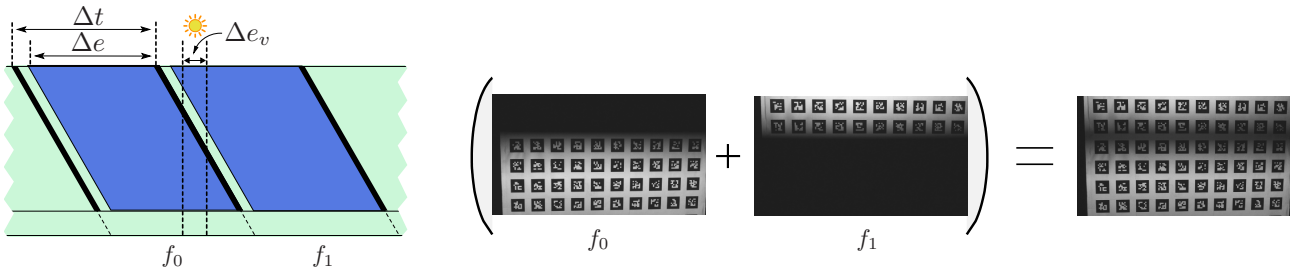

Figure 4. A non-continuous camera exposure results in scanlines with less than full intensity after the summation.

identifying the first synchronized frame. With this approach we can robustly synchronize multiple cameras without requiring specific image content such as trackable features or detailed texture.

5. Method 2: Subframe Warping

Our second technique, while being less accurate than the previous, is applicable to more general illumination conditions. It is based on interpolating intermediate frames, using different offsets for each scanline. Given two consecutive recorded frames, I_n and I_{n+1}, the temporal shear can be removed by interpolating or warping between the two frames.

Linear interpolation may work for some scenes, but in general, especially with higher frequency content, better results can be obtained by morphing. We obtain optical flow vectors $\bar{u}(x, y) = (u, v)$ describing the displacement of a pixel (x, y) from frame n to frame $n + 1$. We then warp along these optical flow vectors to create a morphed image M as follows

$$\begin{aligned} M(x,y) &= (1-\alpha)\cdot I_n(x+\alpha u,\, y+\alpha v) \\ &+ \alpha \cdot I_{n+1}\left(x-(1-\alpha)u,\, y-(1-\alpha)v\right), \end{aligned}$$

where $\alpha = y/S$ is a blending weight that varies as a function of the scanline index. This produces a vertical slice through the spatio-temporal volume in Figure 1 at timestep $t^{(n+1)}$. In the case of interlaced video, we compute optical flow between successive fields, after shifting every second field vertically by half a scanline.

There is nothing to prevent us from shifting α by an arbitrary offset δ, which allows for multiple cameras to be synchronized if we know their relative offsets. Finding such an offset is easy if stroboscopes are available. Even in outdoor environments a strobe light can be aimed directly into the camera lens. As described in Section 4 (see Figure 2), the scanline y at which we observe a transition from the block of bright scanlines back to darkness indicates the time at which the strobe was flashed. Assuming we have already naïvely synchronized the cameras to integer field precision, the subfield offset (in seconds) between cameras C_p and C_q is

$$\left(\frac{y_p - y_q}{S}\right)\Delta t \qquad (3)$$

Dividing this by Δt gives the offset $\pm\delta$ (depending on the ordering of cameras). Note that if $\delta \neq 0$ then when computing M, for some scanlines, $|\alpha + \delta|$ will exceed 1. In this case we have stepped across into the next sheared slice of the volume and have to work with I_{n+1} and I_{n+2} (or I_{n-1} and I_n) instead.

If stroboscopes are not available, then temporal offsets can be obtained via other means, such as by filming continuous periodic motion and detecting trajectories [5].

6. Experiments

For our experiments, we use between 1 and 16 Sony HDR-SR7 camcorders. These camcorders follow the 1080i/30 format [8]. That is, video is recorded at 29.97 frames per second (approximately 60 fields per second interlaced), and each frame has a final visible resolution of 1920×1080. Like many consumer devices, the video is recorded in *anamorphic* mode, where the horizontal direction is undersampled by a factor of $4/3$, meaning that each frame is represented as two fields with a resolution of 1440×540. This anamorphic distortion does not affect the algorithms presented in this paper.

For the stroboscope illumination experiments with instantaneous exposure, we use 3 hardware-synchronized Monarch Instrument Nova-Strobe DAX stroboscopes. This model allows very precise flash rates (between 30 and 20,000 flashes per minute), and short flash durations (20-50 μs). For experiments with varying virtual exposure, we use up to 16 Chauvet LED Techno Strobes that we reprogrammed to allow precise, variable flash duration and flash rates. We use multiple spatially distributed strobes instead of just one in order to increase the intensity and uniformity of the illumination.

We now discuss the synchronization and rolling shutter compensation experiments we performed in order to validate our two proposed techniques. We conclude this section with an experiment demonstrating the trade-off between camera noise and motion blur when setting the virtual exposure time via strobe flash duration.

6.1. Synchronization

Stroboscope Illumination. Our first method for synchronization is demonstrated with the classic example of a

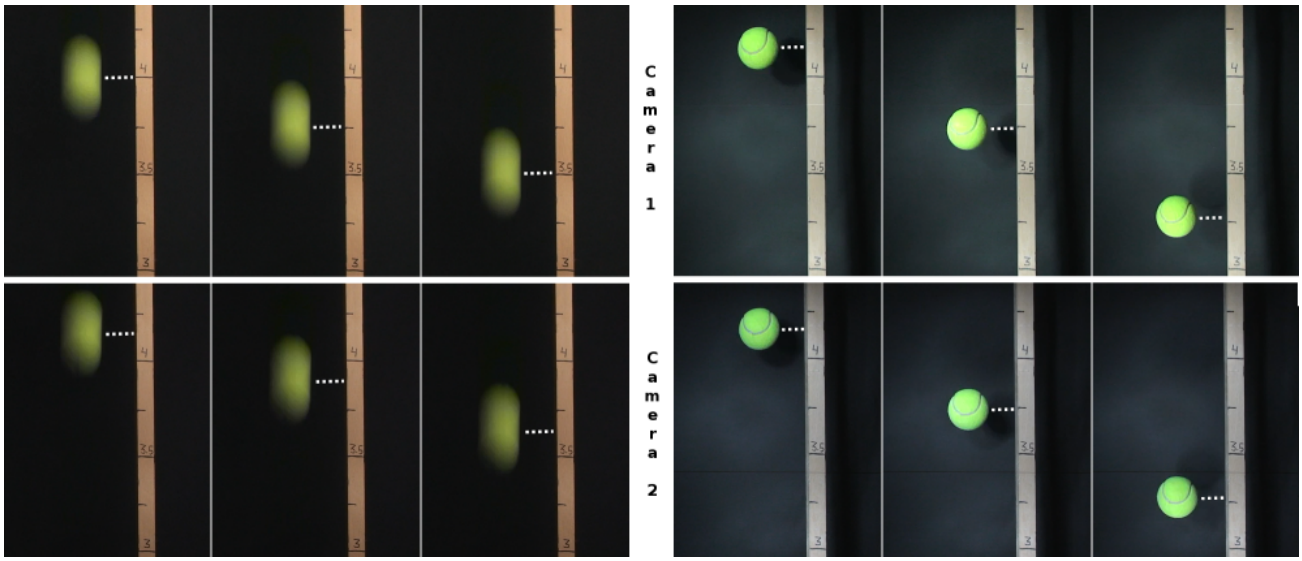

Figure 5. Stroboscope synchronization experiment. Left: 3 consecutive frames (left to right) from 2 unsynchronized cameras (top and bottom). Right: 3 consecutive frames (left to right) from 2 cameras (top and bottom) synchronized by strobe lighting.

falling ball, depicted in Figure 5. Two cameras observe a tennis ball falling beside a measuring stick. On the left side of Figure 5, we show three consecutive frames for each camera, captured with regular, constant illumination. The ball is falling quickly, so the cameras record motion blur. We measure the height of the ball at the center of the blur, as indicated by the dashed white line. It is clear that the cameras are not synchronized. On the right side of the figure, we show the same example using stroboscope illumination. Measuring the height of the ball demonstrates the precise optical synchronization. Note also that this method avoids motion blur, since the frames are captured at instantaneous moments in time. This benefit allows us to accurately capture very fast motions.

Note that the amount of motion between consecutive frames in the synchronized example is roughly twice that of the unsynchronized case, since the strobe frequency was set to half the camera frame rate as discussed in Section 4.

Subframe Warping. In multi-camera setups where stroboscope illumination cannot be used, we can still perform synchronization via the subframe warping method. Figure 6 shows a rotating arrow filmed from two different cameras. A single stroboscope flash was used to obtain the relative time offset.

Assessing Camera Drift. We tested frame rate stability and consistency for 16 consumer cameras of the same model. The cameras were arranged in a half-circle and pointed at a diffuse ball in the center. The ball was illuminated by stroboscope illumination set to the NTSC frame rate of 29.97 Hz. As we discussed in Section 4, flashing at this rate results in a split image where the scanline at which

Figure 6. Subframe warping synchronization. Left: two consecutive overlaid fields from first camera. Center: closest integer frame aligned field from second camera. Right: warped field from first camera, synchronized to match the second.

the split occurs should not move. If either the camera frame rate or the strobe were exhibiting temporal drift, the split scanline would move up or down, indicating a mismatch in illumination and recording frequency. While observing video footage of the 16 cameras recorded for more than 2 minutes, we did not see temporal drift in any camera. Since all cameras were observing the same strobe signal this indicates that frame rates are very stable across different cameras. From this we conclude that the cameras have negligible temporal drift and good frame rate stability for extended periods of time.

6.2. Rolling Shutter Compensation

Any motion orthogonal to the rolling shutter's direction results in a warping of straight lines, and vertical motion results in stretch. Static scenes are obviously unaffected by the rolling shutter, whereas too fast a motion causes blur that somewhat hides the distortion. However, at reasonably quick handheld panning speeds, the distortion can be quite severe, as shown in Figure 7. Since the wall edge covers many scanlines, there is a relatively long time difference

between when the top and bottom of it are captured. Horizontal edges by contrast are not affected to such a large degree. The rotating checkerboard in Figure 8 shows how straight lines are rendered as curves under a rolling shutter.

Figure 7. Original (left) and corrected (right) frame from a hand-held panning sequence. The red line shows how the vertical wall is displaced by as much as 8 pixels in the uncorrected version.

Stroboscope Illumination. Our stroboscope illumination technique completely avoids distortion caused by rolling shutter cameras since all scanlines in a frame are exposed simultaneously by the strobe lighting. This is demonstrated in Figure 8 (bottom right), where the rotating checkerboard is captured with stroboscope illumination. Despite the fast motion, straight lines are captured correctly.

Subframe Warping. Figure 8 also shows our rolling shutter correction results for continuous lighting. Here, we used Horn-Schunck optical flow to compute the warped image. Note that problems can therefore occur in scenes containing occlusions. We were able to correct the fast-moving checkerboard to match the undistorted lines captured in the *stationary* and *stroboscope* cases.

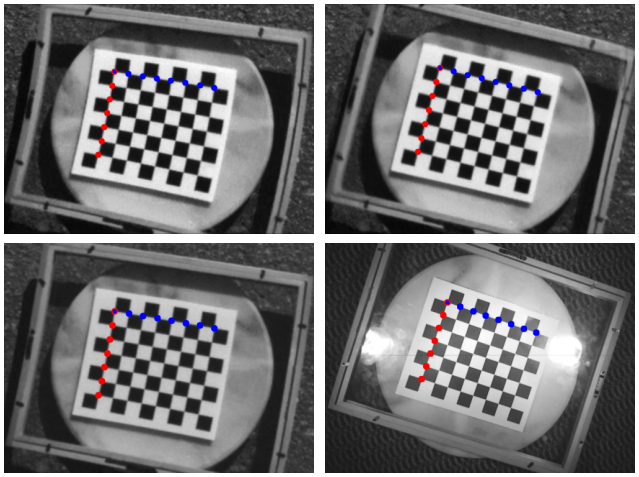

Figure 8. Top left: a fast rotating checkerboard (1 rev. per second) in which straight lines are warped into curves. Top right: after rolling shutter correction by subframe warping. Bottom left: slower motion (half the speed) still produces noticeable distortion. Bottom right: the same scene captured with strobe illumination.

Table 1 contains the residual vector error (L_2) as well as the worst case perpendicular distance (L_∞) for the (approximately) vertical and horizontal rows of indicated corner points to straight line fits. As we can see, both approaches effectively remove the rolling shutter distortion.

	Vertical line		Horizontal line	
	L_∞	L_2	L_∞	L_2
Fast-moving	1.38	2.55	0.19	0.29
Slow-moving	0.57	0.98	0.20	0.37
Stationary	0.10	0.15	0.11	0.19
Subframe warp	0.08	0.14	0.14	0.28
Stroboscope	0.10	0.18	0.09	0.17

Table 1. Norms of perpendicular distance error vectors for the indicated corner points to straight line fits. The vertical line (red), which crosses many scanlines, is more distorted than the horizontal (blue) line.

6.3. Camera Noise vs. Motion Blur

As we discussed in Section 4, the virtual exposure time can be varied to trade off camera noise for motion blur, while keeping the camera exposure constant. We show this effect in Figure 9, with a scene consisting of a stationary checkerboard and a fast rotating ball with constant speed. On the left is a frame captured with a short virtual exposure (0.3 ms), and on the right is a similar frame captured with a longer virtual exposure (6 ms). For the short exposure, we set the camera gain to match the same overall intensity of the longer exposure.

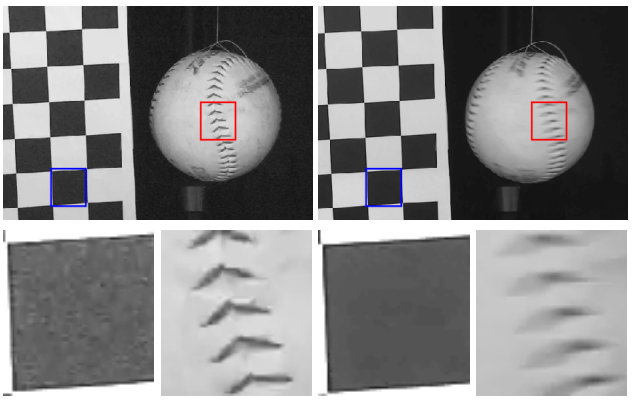

Figure 9. Noise vs. motion blur trade-off. A ball with fast constant rotation and a stationary checkerboard. Top left: short virtual exposure. Top right: long virtual exposure. Bottom row: zoom regions show noise in the short virtual exposure and motion blur in the long virtual exposure. The noise in the zoom regions is amplified by a factor of 2 for better visualization.

The zoom regions at the bottom of Figure 9 show the noise and motion blur difference between the two different virtual exposures, highlighting the trade-off.

7. Discussion and Conclusion

We describe a model for spatio-temporal distortions in video sequences generated by rolling shutter cameras, and present two methods to compensate for these distortions as well as synchronize multiple cameras. The first method uses stroboscopic illumination. This method requires active illumination, therefore limiting its applicability to relatively small-scale indoor environments. The approach also results in a loss of frame rate. On the other hand, the method is very easy to set up, and the post-processing can easily be performed in real-time. The method is therefore ideally suited for on-the-fly processing of video streams. Since the stroboscope can eliminate motion blur, the method can deal with very fast motion. With a controllable virtual exposure time, we allow a trade-off between motion blur and camera noise.

The second method, on the other hand, requires computationally more expensive optical-flow-based warping, and is thus best suited for offline post-processing. As is typical for optical flow methods, this approach can fail when the scene has little high frequency structure or excessive motion, and it can result in distortions and other artifacts near occlusion boundaries. In practice, such artifacts primarily affect background objects, which are usually less interesting for computer vision applications. The key advantage of this approach is that it does not require active lighting, and is thus ideal for outdoor and other large-scale environments. No additional hardware is required beyond the camera(s).

We have used the two methods presented in this paper in several capture projects, including time-resolved reconstruction of non-stationary gas flows [2], multi-view stereo reconstruction [3] of video sequences, and time-varying deformable object capture similar to Bradley et al. [4]. Together, the two approaches for rolling shutter compensation and synchronization enable the use of rolling shutter cameras in a large range of computer vision applications. With these issues resolved, we believe that consumer camcorders are very attractive solutions for computer vision settings due to their low cost, guaranteed frame rate, and easy handling.

References

[1] O. Ait-Aider, A. Bartoli, and N. Andreff. Kinematics from lines in a single rolling shutter image. In *Proc. of CVPR*, pages 1–6, 2007.

[2] B. Atcheson, I. Ihrke, W. Heidrich, A. Tevs, D. Bradley, M. Magnor, and H.-P. Seidel. Time-resolved 3d capture of non-stationary gas flows. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 27(5):132, 2008.

[3] D. Bradley, T. Boubekeur, and W. Heidrich. Accurate multi-view reconstruction using robust binocular stereo and surface meshing. In *Proc. of CVPR*, 2008.

[4] D. Bradley, T. Popa, A. Sheffer, W. Heidrich, and T. Boubekeur. Markerless garment capture. *ACM Trans. Graphics (Proc. SIGGRAPH)*, 27(3):99, 2008.

[5] R. L. Carceroni, F. L. C. Padua, G. A. M. R. Santos, and K. Kutulakos. Linear sequence-to-sequence alignment. In *Proc. of CVPR*, pages 746–753, 2004.

[6] Y. Caspi, D. Simakov, and M. Irani. Feature-based sequence-to-sequence matching. *IJCV*, 68(1):53–64, 2006.

[7] C. Dai, Y. Zheng, and X. Li. Subframe video synchronization via 3D phase correlation. In *Proc. of ICIP*, pages 501–504, 2006.

[8] ITU. ITU-R BT.709, Basic parameter values for the HDTV standard for the studio and for international programme exchange. Standard Recommendation 709, International Telecommunication Union, 1990.

[9] ITU. ITU-R BT.601, Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios. Standard Recommendation 601, International Telecommunication Union, 2007.

[10] C. Lei and Y.-H. Yang. Tri-focal tensor-based multiple video synchronization with subframe optimization. *IEEE Transactions on Image Processing*, 15(9):2473–2480, 2006.

[11] C.-K. Liang, L.-W. Chang, and H. H. Chen. Analysis and compensation of rolling shutter effect. *IEEE Transactions on Image Processing*, 17(8):1323–1330, 2008.

[12] C.-K. Liang, Y.-C. Peng, and H. H. Chen. Rolling shutter distortion correction. In *Proc. of VCIP*, pages 1315–1322, 2005.

[13] C. Linz, T. Stich, and M. Magnor. High-speed motion analysis with multi-exposure images. In *Proc. of Vision, Modeling, and Visualization*, pages 273–281, 10 2008.

[14] M. Meingast, C. Geyer, and S. Sastry. Geometric models of rolling-shutter cameras. In *Proc. of Int. Workshop on Omnidirectional Vision, Camera Networks, and Non-classical Cameras*, 2005.

[15] S. P. Nicklin, R. D. Fisher, and R. H. Middleton. Rolling shutter image compensation. In *RoboCup 2006*, pages 402–409, 2007.

[16] P. Shrestha, H. Weda, M. Barbieri, and D. Sekulovski. Synchronization of multiple video recordings based on still camera flashes. In *Proc. of ACM Multimedia*, pages 137–140, 2006.

[17] S. Sinha and M. Pollefeys. Synchronization and calibration of camera networks from silhouettes. In *Proc. of ICPR*, pages 1:116–119, 2004.

[18] G. P. Stein. Tracking from multiple view points: Self-calibration of space and time. In *DARPA IU Workshop*, pages 1037–1042, 1998.

[19] C. Theobalt, I. Albrecht, J. Haber, M. Magnor, and H.-P. Seidel. Pitching a baseball – tracking high-speed motion with multi-exposure images. In *Proc. of SIGGRAPH*, pages 540–547. ACM SIGGRAPH, 2004.

[20] H. Wang and R. Yang. Towards space-time light field rendering. In *Proc. of I3D*, pages 125–132, 2005.

[21] M. Wäny and G. P. Israel. Cmos image sensor with nmos-only global shutter and enhanced responsivity. *IEEE Transactions on Electronic Devices*, 50(1):57–62, 2003.

[22] B. Wilburn, N. Joshi, V. Vaish, M. Levoy, and M. Horowitz. High-speed videography using a dense camera array. In *Proc. of CVPR*, volume 2, pages 294–301, 2004.