Improved Low-Light Camera Performance Using Multiple Captures

Aaron Deever and Efraín Morales; Eastman Kodak Company; Rochester, NY

Abstract

Current digital cameras suffer from poor performance in low-light situations when a flash is either not available or not beneficial. In these conditions, the signal-to-noise ratio is low, and digitally boosting the signal strength simultaneously boosts the noise to unacceptable levels. In this paper, we present an algorithm that utilizes multiple low-light captures to generate an improved output image. Global and local motion estimation techniques are proposed to align the multiple captures, followed by temporal filtering to achieve noise reduction. Regions of unmatched local motion are detected and excluded from the filtering process. Adaptive spatial filtering is also proposed to further reduce noise.

Introduction

Current digital cameras suffer from poor performance in low-light situations when a flash is either not available or not beneficial. Exposure time can be increased to boost the number of photons reaching the sensor, but this solution typically reduces sharpness in the image if there is any motion in the scene or if the camera is not held absolutely steady. Digital cameras can also artificially boost the light intensity with a digital gain factor. The gain factor effectively scales upward the output codevalue for each pixel. The problem with this technique is that it amplifies noise as well as signal content. Low-light images typically have low signal-to-noise ratios (SNR), and the gain factor required to boost the images to acceptable light levels also causes unacceptable noise levels to be present in the images as well.

In this paper, we address the problem of improving low-light camera performance by using multiple captures. Scenarios in which a flash is not available, such as in many camera phones, or not beneficial, such as many outdoor shots, are considered. A sequence of high-resolution images is captured and combined to produce an improved single output image.

Given a sequence of low-light captures, the problem of producing an improved output image can be approached using temporal noise reduction techniques. The available extra images can be used to reduce the noise present in the target image. If the scene remains completely static across all the images, then temporal noise reduction can be achieved simply by averaging the data temporally. In this ideal case, each pixel location has available several samples of the same pixel value, differing only by an independent noise component in each image. In practice, however, there are several difficulties that must be addressed. Slight camera jitter between captures results in offsets between images. Even if this global motion is accounted for, any local motion within a scene must also be detected. This can be particularly difficult in low-light regions in which the SNR is no greater than one (the noise is as strong as the signal content).

In this paper, we propose an algorithm for achieving improved low-light performance in the previously described scenario. A fast global motion estimation step to correct for camera jitter is followed by a local motion refinement that detects and corrects for object motion. Temporal sigma filtering is performed for regions that have been successfully motion-compensated. Finally, an additional adaptive spatial sigma filtering step is proposed.

The remainder of this paper is organized as follows. In the next section, the proposed algorithm is presented. Experiments and results are subsequently detailed, and a summary completes the paper.

The Algorithm

Temporal noise reduction has been researched thoroughly and is well documented in a review paper by Brailean et al. [1]. To minimize loss of sharpness, it is important to make sure that corresponding pixels from pairs of images are registered properly. This is often achieved through motion compensation, which is used to align each additional frame with the target reference frame. In this paper, a two-step motion estimation and compensation model is proposed. In the first step, a fast, global translational motion estimate is calculated using the technique of integral projections [2,3]. Some local inaccuracies in this motion estimate may remain, requiring a second refinement step. These local inaccuracies may be due to camera nonlinearities, such as barrel distortion. Or they may be due to the inadequacy of a translational motion model to represent more complex motion (e.g., rotation or perspective changes). Or they may be due to local motion within the scene. To account for these errors, the second step of the motion estimation algorithm comprises a local block-based estimate using a restricted search range. The restricted search range avoids excessive computational complexity, and relies on a good initial estimate from the integral projection technique.

Once the frames are aligned, the data can be combined to produce a target image with reduced noise. Simple averaging is not sufficient, however, as this blurs edges and reduces sharpness. A sigma filter is often used in temporal filtering, however, it requires that the SNR be greater than one in order to perform well [4]. Otherwise, a threshold can not be set to reduce noise without blurring signal content in areas where the alignment is not perfect. To solve this problem, a local block-based error metric is used to classify each block as successfully or unsuccessfully motion-compensated. Only those blocks that are successfully motion-compensated are included in the temporal sigma filtering process.

At the completion of temporal filtering, a final spatial sigma filter is used to adaptively smooth more aggressively those regions of the target image that had few successful motion compensation matches from other images, and that therefore had minimal noise reduction achieved in the temporal filtering process.

Global Motion Estimation

Integral projections are used to obtain a fast, robust estimate of the dominant global translational motion between two frames. Local variations, as described above, can subsequently be identified in a second block-based motion refinement step. Details on integral projections are provided in [2,3]. Briefly, either luminance or green channel data is sufficient for the estimation. The data is typically subsampled as well, so green pixels from Bayer pattern data can be used successfully [5]. Data from a two-dimensional image is reduced to two one-dimensional vectors by summing data in each row to form a horizontal projection vector, where each element of the vector represents the sum of the corresponding row of image data, and similarly summing data in each column to form a vertical projection vector, as illustrated in Figure 1. This process is repeated for a second image as well.

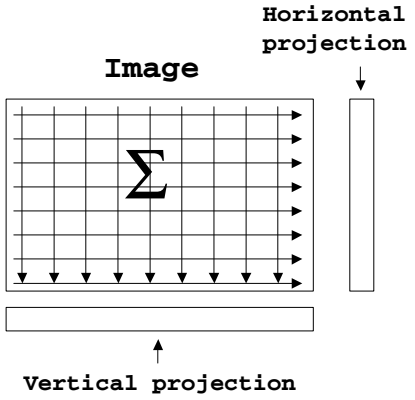

***Figure 1**. Integral projections. The two-dimensional image is converted into two one-dimensional projection vectors by summing along rows, and down columns.*

The vertical projection vectors from the two images are correlated to find the offset providing the best match. Typically an L_1-norm (sum of absolute differences) is used as the error metric, and the offset with the lowest error is chosen as the horizontal motion between the two frames. The process is repeated independently with the horizontal projection vectors to determine a vertical motion estimate.

Computational speedups are possible using two varieties of subsampling. The first subsampling reduces the number of samples included when summing a given row or column. Usually it is desirable to have at least 100 samples included in each sum if possible. Excessive subsampling can result in aliasing that decreases the accuracy of the motion estimate. The second subsampling involves reducing the precision of the motion estimation by only summing data for a subset of the rows or columns. Some precision can be reacquired by interpolating the derived projection vectors prior to correlating them at various offsets. Normally this subsampling is restricted to a factor of two, which is recovered by interpolation of the projection vectors.

Local Motion Estimation

For several reasons outlined above, a global translational motion estimate may not be sufficient to accurately represent the relationship between two frames. In order to allow local corrections of the global motion estimate, a block-based refinement step is used. Block-based motion estimation adds significant complexity to the overall algorithm, so it is desirable to restrict the search range during the matching process to minimize this cost. In this work, we use a search range of ±4 pixels, using Bayer pattern data restricted to offsets of multiples of two (so that the Bayer patterns align). Furthermore, only the green channel Bayer data is used to determine the best offset.

Temporal Sigma Filtering

After aligning the frames, the data is combined to achieve the desired noise reduction. Ideally, all the data is averaged together to maximize noise reduction. In practice, however, averaging of slightly misaligned data results in a loss of sharpness at edges, and can cause ghosting artifacts when motion within the scene is not adequately compensated. To prevent large temporal filtering errors, temporal sigma filtering is used to restrict the averaging process to include only those pixel values within a certain threshold value of the reference pixel value.

For low-light captures, however, there remains a significant hurdle to achieve successful temporal filtering. Sigma filtering requires signals with a high SNR to be successful. The sigma filtering threshold is designed to include in the average those pixels that differ from the reference pixel only by noise while excluding pixels corresponding to different image structure (e.g., across an edge). SNRs for low-light captures are often very low, however, and a threshold set to a typical level of 2σ, where σ is the standard deviation of the noise, may result in varying image structure being included in the temporal average. Reducing the threshold to prevent inclusion of varying image structure decreases the amount of noise reduction possible as well.

On a pixel level, it is very difficult in low-light captures to determine whether the variation from one frame to the next is caused by noise or by different image structure (from slight misalignment or complete motion compensation failure). At the block level, however, it is possible to recognize an unreliable match and to exclude this data from the temporal sigma filtering operation.

Analyzing the L_1-norm cost of the best match for a given block is not sufficient to detect poor matches in many cases. When the SNR is low, the L_1-norm cost resulting from just noise variations may be very similar to the L_1-norm cost associated with a block containing different image structure. Using the L_1-norm, it is very difficult to set a robust threshold that identifies when block matches can be trusted to be good matches.

A better method of detecting poor block matches in low-light captures exploits the fact that in a good match, the corresponding pixels will vary only with a zero-mean noise component. When varying image structure is present, it is likely that the matching error will not be zero-mean. (The mean *absolute* error may be similar in the two cases, but the mean error is typically smaller for blocks differing only by noise.)

Given the noise standard deviation for a single pixel, the noise standard deviation for an entire block can be computed, and a multiple of this term used as a threshold to detect blocks whose best match is likely to contain significant image structure variation. Such blocks are excluded from temporal sigma filtering. In aggregate, these block exclusion decisions form a binary map.

The block exclusion map for a given image can be made more robust by applying general morphological operations to eliminate likely outliers and widen the border of suspected motion regions as a conservative measure. In particular, any block tagged for exclusion that has all immediate eight-neighbors included can be switched to inclusion status. Similarly, any block tagged for inclusion that has at least four of its immediate eight-neighbors tagged for exclusion can be switched to exclusion status. This second step helps to guarantee that the entirety of moving regions is excluded.

This technique is very successful at reducing ghosting artifacts that occur when the temporal filtering step includes data from a moving region that is not successfully motion-compensated.

Spatial Sigma Filtering

Spatial sigma filtering is commonly applied to the problem of noise reduction for a single image. In the proposed low-light scenario, temporal noise reduction is used initially to exploit the presence of multiple frames and reduce the noise present in the reference image. Once this step is completed, standard spatial sigma filtering can be applied, with the variation that the temporal filtering has changed the noise statistics of the image in a potentially nonuniform way.

In a static scene in which every pixel is registered correctly and all data from multiple frames is used in the temporal average, the standard deviation of the noise is reduced by a factor of $\sqrt{n}$, where n is the number of images. If blocks of data are excluded from the averaging filter for some regions, the expected reduction in noise becomes nonuniform, and it is necessary to make the spatial sigma filter adaptive to the changing standard deviation of the noise that is a function of the number of data values included in the temporal filter.

Nonuniformity of the temporal noise reduction may also result in visible noise variation in the output image. To minimize this effect, the spatial sigma filtering neighborhood is also made adaptive as a function of the number of data values included in the temporal filter. A larger spatial neighborhood is used for pixels that achieved less temporal noise reduction. The larger neighborhood increases the potential for noise reduction for these pixels, but comes with the cost of loss of sharpness.

Experiments and Results

Simulations were performed in MATLAB using raw sensor data obtained from a Kodak EasyShare CX7430 zoom digital camera. Burst mode was used to collect six frames of data at a rate of approximately three frames per second. Each image was captured as Bayer pattern data, and temporal noise reduction was performed on the Bayer data. The use of Bayer data for temporal noise reduction avoided the CPU time and memory needed to process and CFA interpolate all six images. Temporal noise reduction on Bayer data also required only approximately one-third the computations of temporal noise reduction on interpolated data. The main drawback of working with Bayer data was that the global and local motion estimation algorithms were restricted to offsets that were multiples of 2, both horizontally and vertically, to ensure Bayer pattern alignment. This resulted in some misalignment and blurring across edges that could potentially be reduced if single pixel motion estimation were used in conjunction with full resolution data.

Six burst images were captured of a scene with a static background and a moving person in the middle of the frame. The camera was hand-held, resulting in slight camera jitter. Figure 2 shows a small version of the original captured reference frame.

***Figure 2**. Original image captured in low light.*

Figure 3 shows small versions of all six images after the gain factor was applied. The reference image is the middle image on the right side. While the background is static for all six images, the person on the couch is moving from frame to frame.

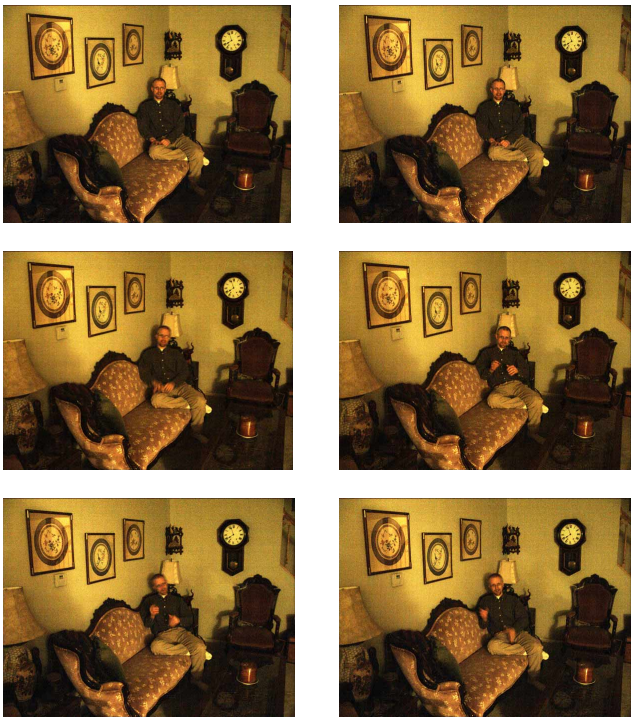

***Figure 3**. Six images with only gain factor applied. The reference frame in this example is the middle image in the right column.*

Figure 4 illustrates the difficulty with temporal sigma filtering of low-SNR images using only global motion estimation. In this case, the integral projection algorithm correctly determined the dominant global camera jitter, but did not correct for local motion within the scene. Because the images were low-light with low SNR values, the temporal sigma filter was unable to distinguish between noise and genuine signal structure difference when filtering. As a result, ghosting artifacts appeared in the moving regions.

***Figure 4**. Global motion estimation followed by temporal sigma filtering. Because of the low SNR of the signals, the sigma filter was unable to differentiate noise from different image content caused by motion.*

The previous result was improved by including a second local motion estimation and block exclusion step, using blocks of size 32×32. This result is shown in Figure 5.

***Figure 5**. Global motion estimation followed by local motion estimation refinement. Matching blocks with too high an error were excluded from the temporal sigma filter.*

As can be seen, the ghosting artifacts were corrected by incorporating the local motion estimation that excluded blocks that were classified as poor matches. The region surrounding the person's head remained very noisy, a result of the fact that few if any frames provided successful block matches in that region. In cases where no frames provided a good match, the resulting region was equivalent to the original reference image region. Figure 6 shows a block exclusion map that represents the number of frames with successful block matches for a particular region. Black regions imply that no matches were found (only the reference image was included in the temporal sigma filter). White regions imply that all six images contributed data to the sigma filter.

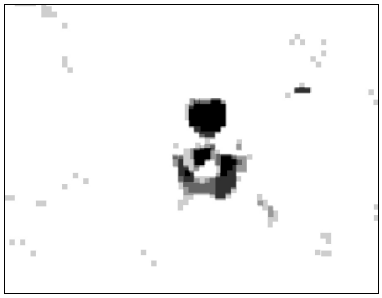

***Figure 6**. Map representing contributions to the temporal sigma filter. Most of the background regions used data from all six images (white regions) while some blocks near the person's hands and head used only the reference image and, therefore, achieved no noise reduction at all.*

A final improvement (Figure 7) was achieved by including a spatial sigma filter step. Regions in which either five or six images contributed to the temporal sigma filter used a 3×3 spatial sigma filter. Regions in which either three or four images contributed to the temporal filter used a 5×5 spatial sigma filter. Regions for which only one or two images contributed to the temporal sigma filter used a 7×7 spatial sigma filter.

***Figure 7**. Noise reduction using both temporal filtering and spatial sigma filtering to blend regions that did not benefit from temporal filtering.*

The spatial sigma filter provided a moderate additional noise reduction benefit, and achieved some success at blending between regions with different amounts of temporal noise reduction, at a cost of decreased sharpness. Significant noise remained in the person's hands and head, along with considerable blurring.

Summary

In this paper we address the problem of improving low-light camera performance by using multiple captures. Global and local motion estimation techniques are proposed to align the images, followed by temporal filtering to achieve noise reduction. An algorithm is proposed to detect regions of unmatched local motion, and exclude them from temporal averaging. Adaptive spatial filtering is also proposed to further smooth the resulting image.

References

[1] J. Brailean et al., Proceedings of the IEEE, 83(9), pg. 1272 (1995).

[2] K. Sauer and B. Schwartz, IEEE Transactions on Circuits and Systems for Video Technology, 6(5), pg. 513 (1996).

[3] K. Ratakonda, IEEE International Symposium on Circuits and Systems, 4, pg. 69 (1998).

[4] J.S. Lee, Computer Vision, Graphics and Image Processing, 24, pg. 255 (1983).

[5] B. Bayer, "Color Imaging Array," US Patent 3,971,065 (1976).

Author Biography

Aaron Deever received his B.S. degree in mathematics and computer science from The Pennsylvania State University, and a Ph.D. degree in applied mathematics from Cornell University. He is currently a Research Scientist at Eastman Kodak Company, Rochester, NY, where he works on image and video processing applications.