Preserving natural scene lighting by strobe-lit video

Citation

Year
2015

Version
Peer reviewed version (post-print)

Link to publication
TUTCRIS Portal (http://www.tut.fi/tutcris)

Published in
Image Processing: Algorithms and Systems XIII

DOI
10.1117/12.2185013

Copyright
Copyright 2015 Society of Photo Optical Instrumentation Engineers. One print or electronic copy may be made for personal use only. Systematic reproduction and distribution, duplication of any material in this paper for a fee or for commercial purposes, or modification of the content of the paper are prohibited.

Take down policy
If you believe that this document breaches copyright, please contact tutcris@tut.fi, and we will remove access to the work immediately and investigate your claim.
Preserving Natural Scene Lighting by Strobe-lit Video

Olli Suominen, Atanas Gotchev
Department of Signal Processing, Tampere University of Technology
Korkeakoulunkatu 1, 33720 Tampere, Finland

ABSTRACT

Capturing images in low light intensity, and preserving ambient light in such conditions pose significant problems in terms of achievable image quality. Either the sensitivity of the sensor must be increased, filling the resulting image with noise, or the scene must be lit with artificial light, destroying the aesthetic quality of the image. While the issue has been previously tackled for still imagery using cross-bilateral filtering, the same problem exists in capturing video. We propose a method of illuminating the scene with a strobe light synchronized to every other frame captured by the camera, and merging the information from consecutive frames alternating between high gain and high intensity lighting. The motion between the frames is compensated using motion estimation based on block matching between strobe-illuminated frames. The uniform lighting conditions between every other frame make it possible to utilize conventional motion estimation methods, circumventing the image registration challenges faced in fusing flash/non-flash pairs from non-stationary images. The results of the proposed method are shown to closely resemble those computed using the same filter based on reference images captured at perfect camera alignment. The method can be applied starting from a simple set of three frames to video streams of arbitrary lengths with the only requirements being sufficiently accurate syncing between the imaging device and the lighting unit, and the capability to switch states (sensor gain high/low, illumination on/off) fast enough.

Keywords: computational photography, image denoising, ambient light, low-light, motion compensation, bilateral filter

1. INTRODUCTION

Ambient light is significant part of the visual experience of a captured scene in terms of aesthetics. However, ambient light in many cases does not provide enough illumination to capture the scene with high quality. For instance, candlelight creates a certain kind of atmosphere that should ideally be preserved when capturing the scene, but is not able to produce enough light for photographic needs. The problem of capturing images in low light has been tackled by numerous approaches. A direct increase of the sensor’s sensitivity would fill the resulting image with noise, therefore, a simultaneous improvement of its noise characteristics must be sought. This is a field of research completely different than computational photography and image processing and as such, we do not explore it here. Another approach relies on prolonging the exposure time. This however increases the risk of introducing motion blur in the video. Alternatively, the scene can be lit with artificial light, which would interfere with the aesthetic properties of the image.

A prominent computational imaging method is to employ flash/non-flash image pair photography⁴, where images taken in two different lighting conditions are merged using cross-bilateral filtering to create a single, high quality image. The concept of fusing two differently illuminated images has since been adopted in several approaches⁵,⁶,⁷. In some cases, (near) infrared light has been used as a non-obtrusive lighting method. The idea has some limitations though, as compared to using simultaneous infrared flashes, the conventional flash better preserves color information, preventing color bleeding which can occur due to lack of notable contrast between colors in the infrared domain⁵. Flashes emitting visible light are also much more commonplace in existing hardware than those only operating at (N)IR wavelengths. Another proposed approach is to use a prolonged exposure time to collect enough light, thus motion blurring the image, and then to use the flash image as a constraint in the optimization to deblur it⁶.

We review the flash/non-flash photography approach in more details, as it is the basis of our proposed method. In 2004, Petschnigg et al. introduced the concept of fusing photographs pairs taken with and without flash⁴. The highly cited paper describes many image enhancements that can be performed on an aligned pair of flash/no-flash images, among them
ambient image denoising. This concept is able to effectively remove the noise in a low-light image by utilizing a high signal-to-noise ratio image of the same scene taken with artificial light to guide the denoising process. However, the input images must be well aligned before filtering, prompting either the use of a tripod or some image registration between the differently illuminated frames, and leading to poor performance if any motion is involved between frames.

The method relies on bilateral filtering, which provides edge-preserving smoothing. The conventional bilateral filter blurs together pixels with similar intensity values in a spatial neighborhood. A spatial low-pass filter acts together with an edge-stopping kernel, which prevents the filter from averaging over areas too different. The response for a single pixel of the bilateral filter can be formulated as

$$Y(x) = \frac{1}{k(x)} \sum_{p \in W} g_d(x - p) g_r(X(x) - X(p)) X(p),$$

(1)

where $W$ is the spatial window around pixel coordinate $x$ and $k(x)$ is the normalization term,

$$k(x) = \sum_{p \in W} g_d(x - p) g_r(X(x) - X(p)).$$

(2)

The functions $g_d$ and $g_r$ define the weights for proximity in spatial and range (intensity) domains respectively. Commonly they are set to Gaussians with respective standard deviations $\sigma_d$ and $\sigma_r$. In this way, the filter adapts the weights based on the input image $X$. However, as the objective is to utilize the high SNR information from the flash image $F$, the filter should be adapted accordingly. This is done by computing the edge preserving function $g_r$ based on $F$ while still applying it to the target image $X$. Thus, the filter becomes

$$Y_F(x) = \frac{1}{k(x)} \sum_{p \in W} g_d(x - p) g_r(F(x) - F(p)) X(p),$$

(3)

which the authors dub the joint bilateral filter. The same modification is done to the normalization term $k(x)$.

Furthermore, the work considers falling back to the conventional bilateral filter in areas where flash shadows and specular reflections may cause under-blurring due to introducing non-existent edges and thus triggering the edge-preservation function in wrong places. Thus, the end result will be a blend of the flash assisted and plain responses $Y_F$ and $Y$. The article goes on to present other types of image enhancement achievable with the same setup, but as our target is expressly in preserving the ambient lighting, those further contributions are not described here. That said, all of the techniques from the original source would be applicable after performing our proposed motion compensated frame reconstruction.

2. PROPOSED CAPTURING SETUP

The imaging setup we consider in this article consists of a conventional video camera and a strobe light. The strobe is synchronized to the imaging speed of the camera such that each other frame captured by the camera is lit with a light pulse from the strobe, and every other captures the ambient light in the scene. For those frames lit by the strobe, the camera sensor is set to ‘normal’ sensitivity (i.e. gain, or ISO rating as more commonly known in photography). For the non-flash frames, the sensor is set to high sensitivity. Exposure time, aperture and other settings are kept constant for both types of frames. Thus, the flash frames capture a clear image with visible edges, texture and detail, but due to the artificial lighting (artificial in the sense that the strobe lighting is not natively a part of the scene), lose much of the original color information in the scene. The non-flash frames, however, capture the color more closely as seen in the scene without disturbance from the strobe. The downside of it is that the high gain required to magnify the lower amount of collected light also magnifies the noise, thus making the non-flash frames correctly colored but noisy. The only specific requirement for the camera is that it should be able to switch between two states of sensitivity (i.e. gain) at the capture frame rate. Another hardware requirement comes from the synchronization of the illumination to the camera acquisition rate.

A lower limit for the exposure time is set by the framerate of the camera. In order to allow motion to appear smooth, the camera must capture a sufficient number of frames, and depending on the setting, this is usually considered to be in the range of 24 to 30 frames per second (FPS). For the purpose of this paper, we consider 30 as the target FPS, however this can be adjusted as necessary. Depending on how the later processing stages are done, the camera in our scenario should
run at double the target FPS (each flash/non-flash pair is combined to a single frame in a 2:1 ratio). The same target frame rate can also be achieved through frame interpolation between the resulting frames, which would make it possible to run the camera at the target FPS. Further consideration should be given to the speed of motion in the scene – too long exposure times will induce motion blur into the frames. While compensating this through computational means is possible, it is outside our scope and we only state that conventional expertise in setting the exposure time in video capture applies here.

The strobe should be capable of being triggered accurately enough to hit every other frame exposed by the camera. The length of an exposure in a low-light setting is in the order of 10-30 milliseconds per frame, while strobes operate and sync in the scale of some microseconds. Therefore the timing requirements are not very demanding, and synchronization should be perfectly doable even if direct access to the camera acquisition trigger is not available. The strobe should be powerful enough to be able to light up the field of view of the camera in a similar fashion to a flash device of a photographic camera. The exact properties of the strobe depend on the particular situation, scene and desired effect. The requirements dictated by our capture setup are that it should be able to periodically provide short, impulse-like flashes at half of the camera frame rate, and trigger accurately enough that it can be synchronized with the frame capture. The system will work as long as the duration of flashes and the synchronization accuracy together allow the illumination to happen within a single frame capture (i.e. during a single exposure time) without spilling over to the adjacent frames.

### 3. MOTION COMPENSATION VIA BLOCK MATCHING

In order for the cross-bilateral filter to function properly, the reference image has to be precisely aligned with the filtering target. As in our application the images are extracted from a video stream, the images are most likely displaced in one way or another. Either camera motion or dynamic objects in the scene have to be considered. An additional problem encountered by earlier works on non-static shots is the difference in properties of flash and non-flash images. It makes the task of registering and aligning the images a much harder one, as the same image areas can look drastically different in the presence of noise and different illumination. We propose to solve this issue by applying the matching to frames captured with the same sensitivity and illumination (Figure 1), and interpolating the motion for the in-between frames.

![Diagram of the processing involved in creating the low noise, ambient lighting frames. Virtual flash frames are constructed by interpolating the motion between two true flash frames, which are then used to denoise the corresponding no-flash frame. Optionally, more sophisticated frame interpolation methods may be used for the resulting stream if halving the temporal sampling rate from the input stream is not acceptable.](image)
Block matching is a well-known method for tracking similarities between images, such as consecutive frames of a video for video compression, or a stereo pair in depth estimation. While many changes and improvements have been suggested to increase its robustness, speed and other properties, the main concept remains the same. The image is divided into reference blocks (overlapping or not, depending on the use case). Each block is then compared to image blocks within a search window in the other image, and a cost metric is applied to each pairing. The cost indicates how closely that particular block resembles the reference block. Common cost metrics are for instance L1 (sum of absolute differences, SAD) and L2 (sum of squared differences, SSD) norms. The block with the lowest matching cost is then selected as the best match for that reference block, and its relative position to the reference block is designated as the block’s motion vector.

### 3.1 Reconstructing the flash reference image

Once matching has been performed between two consecutive flash images, the motion vectors are used to reconstruct a virtual flash image, which aligns with the captured non-flash image in between. Reference blocks are shifted along their motion vectors and aggregated in a single image (Figure 2). A way to do so is to assume all motion in the video follows a linear path during the span of the three frames which are used at a time. The placement for a block in the middle frame is thus along the motion vector $\mathbf{v}$, i.e. $\alpha \mathbf{v}$. Considering the temporal sampling frequency of >30 frames per second, it can be argued that this assumption is reasonable.

![Figure 2. The motion compensated reconstruction of the "missing" flash frame based on motion vectors. The path of motion is assumed to be linear during the three frames, so motion vectors $\mathbf{v}$ are scaled by $\alpha$ to get the new block placements. Reconstruction may leave some areas blank (black areas in frame $n$) due to gaps between the interpolated positions, which will then be filled in before using the image in the bilateral filter.](image)

In order to find the scaling coefficient, registration should be done between the reconstructed flash frame and its non-flash counterpart. As global registration is sufficient, the task is notably easier than the dense mapping we are avoiding by originally using the two flash frames. Even though the non-flash frame is significantly different in almost all aspects, making attempts at color based registration impractical at best, both of the frames do share a significant number of edges. By applying a 3x3 Sobel filtering, we extract the edge information from the non-flash frame $n$ and a set of hypothesis reconstructions $P_{\alpha}^n$, where $\alpha \in [0,1]$. The reconstruction whose edges have the highest correlation (Figure 3) with the non-flash frame is selected as the most suitable.

![Figure 3. The Sobel filtered edges in the non-flash frame $n$ (left), one reconstructed flash image (middle) and their correlation over the motion vector estimated between $n-1$ and $n+1$ (right). Note the gaps left by the reconstruction algorithm, which do not interfere with the registration and can thus be left without filling until the correct one is selected.](image)
The reconstructed frame will have some missing information due to gaps between the shifted blocks (Figure 2). The problem is to some extent reduced by doing the reconstruction twice – backwards and forwards – and averaging the two estimates pixel-wise in places for which more than a single value was attained. For the gaps still remaining, the important thing is to avoid having false edges, which would trigger the edge function. Therefore we fill those with linear interpolation.

3.2 Outliers in block matching

Block matching tends to fail in finding correct correspondences in areas of low texture, which may lead to erroneous reconstruction of the synthetic flash frame. However, in the proposed approach, the reconstructed flash frame is never presented to the user. Instead, it is used as the reference image in the bilateral denoising stage. Therefore it is sufficient to analyze, how the problematic areas and the following wrong matches affect the filtering outcome.

The reference image determines the filter weights applied to the target image in an attempt to preserve edges. If there is an edge in the reference, the weights accommodate this by only blurring the area on the target side of the edge, and giving low weights to the parts which do not have similar intensities. If there is no edge, the filter acts like a conventional spatial filter, blurring the contents of the whole filter window. Essentially, the presence of detail and texture (edges) makes it likely that the block matching will be able to extract a good motion vector for that area, and thus the filter will preserve that detail based on the successful reconstruction. If, however, there is no significant detail, block matching will fail, but neither does the filter have any detail to preserve in the area. The texture dependence of block matching therefore does not pose a serious problem to the effectiveness of the proposed algorithm.

![Figure 4](image)

Figure 4. A few plausible best matches for an area with no texture. While none of them are actually correct, the nearby blocks will not have any influence on the denoising outcome. However, should the furthest one be selected, it will introduce false texture in the reconstructed reference image and may either unnecessarily trigger the edge stopping function, or mask true edges and over-blur the image.

Issues arise, if the motion vector gets wrongly estimated to pass through an area which has a notably different texture. Then the reconstruction step will interpolate the block to be placed in the middle of another texture, potentially masking the true surface and creating false edges. To counter this issue, motion vectors for low texture blocks can be constrained to travel along connected regions only. However, this will impose some restrictions on the maximum allowable speed of motion. An even simpler method, which we found to be working quite robustly in most cases, is to have a small distance penalty in selecting the best matches.

4. EXPERIMENTS

For presenting the results of the proposed approach, we captured data sets of a Lego truck and an array of beverage containers using a Nikon D5200 DSLR, as depicted in Figure 5. In the Lego scene, synthetic light comes from the integrated flash on the camera, while the ambient lighting is from a LED studio light set at approximately 3200K color temperature and its lowest intensity setting. In the container scene, the synthetic light is provided by the aforementioned LED light set at a temperature of 5600K and approximately half of the full intensity, whereas the ambient light comes completely from the two small candles. The camera sensitivity is 100 ISO in the frames with synthetic lighting and 6400 in the ambient frames. In order to capture a “ground truth” flash image for the in-between images for evaluating the relative quality of the motion compensation, a tripod was used to control the camera motion in between shots \(n-1\), \(n\), and \(n+1\). A comparison is presented in Figure 6 with frames made by blending together the two control images exactly aligned by a tripod. The same filter settings have been applied to both image pairs.

*Color images of the input sequences and the results are available in the electronic version of the paper*
Figure 5. Two input sequences of three frames, first and third illuminated with a flash and the second one captured in ambient lighting. Camera pans from left to right in the upper row frames, and translates vertically in the bottom row.

Figure 6. Bilateral filtering results from the input sequences in Figure 1 via motion compensation (left) and with perfectly aligned images taken from the same scene with a tripod (right).
5. CONCLUSIONS

We have presented a motion compensation approach for combining artificially lit image frames with ones taken in ambient lighting, which makes it possible to apply the well-known flash/no-flash fusion method to video with a moving camera and non-static scenes. The processing chain is simple, and utilizes readily available image processing tools. The hardware requirements on the syncing and sensor speed are specific, but reasonable, which makes implementing this setup in practice quite feasible. For instance, a mobile phone with an LED flash may well be enough from a hardware point of view, given that sufficient control over the sensor and the flash can be gained. Although processing is happening in the temporal dimension, the needed amount of frames is low. As only three consecutive frames are required, a promising application is to take a flash/no-flash/flash sequence to get a single, high quality image with preserved ambient lighting. As future work, the robustness of the motion compensation can always be improved. So far, the results have been generated in a simulated laboratory environment, where the camera motion can have different characteristics compared to an actual handheld device. Constructing a prototype device with the proper strobe syncing is naturally the next appropriate step.

REFERENCES