ADAPTIVE BLOCK-BASED APPROACH TO IMAGE STABILIZATION

Marius Tico

Nokia Research Center
955 Page Mill Road, Palo Alto, CA, USA
Email: marius.tico@nokia.com

ABSTRACT

The objective of image stabilization is to prevent image blurring caused by the relative motion between the camera and the scene during the image integration time. In this paper we propose a software approach to image stabilization based on capturing and fusing multiple short exposed image frames of the same scene. Due to their short exposure, the individual frames are noisy, but they are less corrupted by motion blur than it would be a single long exposed frame. The proposed fusion method is designed such that to compensate for the misalignment between the individual frames, and to prevent the blur caused by object motion in front of the camera during the multi-frame image acquisition. Various natural images acquired with camera phones have been used to evaluate the proposed image stabilization system. The results reveal the ability of the system to improve the image quality by simulating longer exposure times. In addition the system has the ability to reduce the effect of noise and outliers present in the individual short exposed frames.

Index Terms— image stabilization, camera motion, de-noising, low light imaging, motion blur

1. INTRODUCTION

Any relative motion between the camera and the scene during the image integration time results in a degradation of the image known as motion blur. This degradation reduces the perceptual quality of the imaging product, and hence the manufacturers prioritize their efforts to develop solutions to this problem. Such solutions are commonly known under the name of ”image stabilization”. The main motivations for such efforts are two-fold. First, it is the need for longer integration times in order to cope with smaller pixel areas that result from sensor miniaturization and resolution increase requirements. Second, it is the difficulty in avoiding unwanted motion during the integration time when using high zoom, and/or small hand-held devices.

Image stabilization solutions developed so far are divided in two categories, based on whether they are aiming to correct or to prevent the motion blur degradation. The first category includes those solutions that are aiming to restore a single image shot captured during a long exposure time. Such image could be affected by motion blur caused by unwanted camera motions during the exposure time. If the blur point spread function (PSF) is known then the original image of the scene can be restored, to some extent, by applying an image deconvolution routine . In such a case, the main limitation in restoring the original image is given by the zeros of the PSF in the frequency domain, that result in lost spatial frequencies from the original image. However, the main difficulty is that in most practical situations the motion blur PSF is not known, and since it depends of the arbitrary camera motion during the exposure time, it is also different for each particular image. The lack of knowledge about the blur PSF suggests the use of blind de-convolution approaches in order to restore the motion blurred images [1, 2]. Unfortunately, most of these methods rely on rather simple motion models, e.g. linear constant speed motion, and hence their potential use in consumer products is rather limited. Measurements of the camera motion during the exposure time could help in estimating the motion blur PSF and eventually to restore the original image of the scene. Such an approach have been introduced in [3], where a secondary video camera is used for estimating the motion during the exposure time of the principal camera. Another way to estimate the PSF has been proposed in [4], where a second image of the scene is taken with a short exposure. Although noisy, the secondary image is unaffected by the motion blur and it can be used as a reference for estimating the motion blur PSF which degraded the principal image. The second category of image stabilization solutions are aiming to prevent the motion blur for happening in the first place. In this category are included all optical image stabilization (OIS) solutions adopted nowadays by many camera manufactures. These solutions are utilizing inertial senors (gyroscopes) in order to measure the camera motion, following then to cancel the effect of this motion by moving either the image sensor or some optical element in the opposite direction. Due to the fact the inertial sensors are less sensitive to low frequency motions, the OIS solutions are effective only for relatively small exposure times. As the exposure time increases the mechanism may drift, producing motion blurred images. A different method, based on specially designed high-speed CMOS sensors has been proposed in [5]. The method utilizes the possibility to independently control the exposure time of each image pixel. In order to prevent motion blur the integration is stopped selectively in those pixels where motion is detected.

Multi-frame image stabilization, is another approach to prevent motion blur, and it relies on dividing a long exposure time in shorter intervals following to capture multiple short exposed image frames of the same scene. Due to their short exposure, the individual frames are corrupted by sensor noises but fortunately they are less affected by motion blur. Consequently, a long exposed and motion blur free picture could be synthesized by registering and fusing the available short exposed image frames. In this paper we introduce a novel and efficient approach to multi-frame image fusion for image stabilization. One input frame is selected as the reference and divided in blocks of variable size in accordance to the image content, i.e. larger blocks in smooth image areas and smaller blocks in the neighborhood of prominent image details. Next, each block of the reference frame is improved based on visually similar blocks found in the available images. It is of importance to mention that the method presented here extends our previous work [6], by (i) allowing multiple correspondences for each image block in both spatial and temporal dimensions, and (ii) adaptively selecting the size of each block in accordance to the image content.

2. THE PROPOSED ALGORITHM

The pixel brightness delivered by an imaging system is related to the exposure time through a non-linear mapping called "radiometric response function", or "camera response function" (CRF). There are a variety of techniques (e.g. [7, 8]) that can be used for CRF estimation. In our work we assume that the CRF function of the imaging system is known, and based on that we can write down the following relation for the pixel brightness value:

$$I(\mathbf{x}) = \text{CRF}\left(g(\mathbf{x})\Delta t\right) \tag{1}$$

where $\mathbf{x} = [x\ y]^T$ denotes the spatial position of an image pixel, $I(\mathbf{x})$ is the brightness value delivered by the system, $g(\mathbf{x})$ denotes the irradiance level caused by the light incident on the pixel $\mathbf{x}$ of the imaging sensor, and Δt stands for the exposure time of the image.

Let I_k, for $k \in \{1, \ldots, K\}$ denote the K observed image frames whose exposure times are denoted by Δt_k. A first step in our algorithm is to convert each image to the linear (irradiance) domain based on knowledge about the CRF function, i.e.

$$g_k(\mathbf{x}) = (1/\Delta t_k)\,\text{CRF}^{-1}\left(I_k(\mathbf{x})\right), \text{ for all } k \in \{1, \ldots, K\}. \tag{2}$$

We assume the following model for the K observed irradiance images:

$$g_k(\mathbf{x}) = f_k(\mathbf{x}) + n_k(\mathbf{x}), \tag{3}$$

where n_k denotes a zero mean additive noise, and f_k denotes the latent image of the scene at the moment the k-th input frame was captured. We emphasize the fact that the scene may change between the moments when different input frames are captured. Such changes could be caused by unwanted motion of the camera and/or by the motion of different objects in the scene. Consequently, the algorithm can provide a number of K different estimates of the latent scene image, each of them corresponding to a different reference moment.

In the following, we assume that g_r, ($r \in \{1, \ldots, K\}$) is the reference observation, and hence the objective of the algorithm is to recover an estimate of the latent scene image at moment r, i.e. $f = f_r$.

The restoration process is carried out based on a spatiotemporal block processing. Assuming a division of g_r in non-overlapping blocks of size $B \times B$ pixels, the restored version of each block is obtained as a weighted average of all $B \times B$ blocks located in a specific search range, inside all observed images.

Let $\mathbf{X}_\mathbf{x}^B$ denote the sub-set of spatial locations included into a block of $B \times B$ pixels centered in the pixel $\mathbf{x}$, i.e.:

$$\mathbf{X}_\mathbf{x}^B = \left\{\mathbf{y} \in \Omega \mid [-B\ \ -B]^T < 2\left(\mathbf{y} - \mathbf{x}\right) \leq [B\ B]^T\right\}, \tag{4}$$

where the inequalities are componentwise, and Ω stands for the image support. Also, let $g(\mathbf{X}_\mathbf{x}^B)$ denote the $B^2 \times 1$ column vector comprising the values of all pixels from an image g that are located inside the block $\mathbf{X}_\mathbf{x}^B$.

The restored image is calculated block by block as follows

$$\hat{f}(\mathbf{X}_\mathbf{x}^B) = \frac{1}{W}\sum_{k=1}^{K}\sum_{\mathbf{y} \in \mathbf{X}_\mathbf{x}^S} w_k\left(\mathbf{x}, \mathbf{y}\right) g_k(\mathbf{X}_\mathbf{y}^B), \tag{5}$$

where $W = \sum_{k=1}^{K} W_k(\mathbf{X}_\mathbf{x}^B)$, with $W_k(\mathbf{X}_\mathbf{x}^B) = \sum_{\mathbf{y} \in \mathbf{X}_\mathbf{x}^S} w_k\left(\mathbf{x}, \mathbf{y}\right)$, denoting the total weight value of the kth image in the block $\mathbf{X}_\mathbf{x}^B$. The set $\mathbf{X}_\mathbf{x}^S$ denotes the spatial search range of size $S \times S$ centered in $\mathbf{x}$, and $w_k\left(\mathbf{x}, \mathbf{y}\right)$ is a scalar weight value assigned to an input block $\mathbf{X}_\mathbf{y}^B$ from image g_k. The weigh values are emphasizing the input blocks that are more similar with the reference block.

In our work we used an exponential like weighted functions of the form

$$w_k\left(\mathbf{x}, \mathbf{y}\right) = \exp\left[-\frac{1}{U^2\sigma_{r,k}^2}\text{dist}\left(g_k(\mathbf{X}_\mathbf{y}^U), g_r(\mathbf{X}_\mathbf{x}^U)\right)\right], \tag{6}$$

where $\sigma_{r,k}^2$ is the sum of noise variances in the reference and the k-th observed image, and $\mathbf{X}_\mathbf{x}^U$ is a block of size $U \times U$, ($U \geq B$), called here *outer-block*, that includes the actual $B \times B$ image block in the middle. Thus, in accordance to (6), the similarity between blocks can be calculated based on matching larger neighborhoods (i.e. outer-blocks) that include the actual image blocks in the middle.

The vectorial distance function "dist", used in (6) is defined as

$$\text{dist}\left(\mathbf{a}, \mathbf{b}\right) = \sum_t \rho\left(a_t - b_t\right), \tag{7}$$

where $\mathbf{a}$, $\mathbf{b}$ are vectors of the same length, and $\rho(u)$ is equal with u^2 if $|u| > \sigma_{r,k}$, and zero otherwise.

The sizes B and U, selected for the image block and outer-block respectively, influence both the computational complexity and the quality of the result. On one hand, a small ratio U/B, i.e. close to one, is desirable in order to reduce the computational cost. On the other hand, a large ratio U/B would be preferable in order to improve the quality of the result. This is because a large outer-block size (U) would ensure a more robust estimate of the block similarity in the presence of noise, whereas a small B would ensure the preservation of sharp edges preventing image over-smoothing.

In our solution the tradeoff between quality and complexity is addressed by adaptively selecting the proper block size suitable for each image region. The selection is based on the estimated restoration quality achievable by different block sizes. Also, in order to reduce the computational load the selection of the block size in different image regions is carried out only into the reference image. The total weight in the reference image (i.e. $W_r(\mathbf{X}_\mathbf{x}^B)$) is used as an estimate of the restoration quality achievable with a block size B in the given image region. Starting with a large block size (i.e. small U/B ratio), the algorithm progressively decreases the block size until the total weight exceeds a certain threshold. Noting that the maximum value of the total weight is S^2, we use a threshold value of the form γS^2, where γ is a real constant that, in accordance to our experimental observations, can be selected in the interval $[0.3, 0.5]$. The propose method selects large block sizes (i.e. small U/B ratios) in the smooth image ares, and small block sizes (i.e. large U/B ratios) only in the neighborhood of image edges, gaining from the fact that edges and sharp transition regions typically represent only a small percent from a natural image area.

The stabilization algorithm receives as parameters a list of $L > 1$ decreasing block sizes, i.e. $B_1 > B_2 > \cdots > B_L$, along with their corresponding outer block sizes, i.e. $U_1 \leq U_2 \leq \cdots \leq U_L$. The appropriate block sizes and block positions determined by the algorithm are stored in a set $\mathcal{B}$ of the form:

$$\mathcal{B} = \{(\mathbf{x}, B) \mid \mathbf{x} \in \Omega,\ B \in \{B_1, \ldots, B_L\}\}, \tag{8}$$

where each element encodes a *block descriptor* comprising the central coordinates of the block ($\mathbf{x}$) and the size of the block (B). The operations applied onto the reference image in order to select the appropriate block sizes for each image region are summarized in the following algorithm:

1. Set $\mathcal{B} = \emptyset$, and an auxiliary set $\mathcal{B}_0 = \emptyset$.

2. Divide g_r in non-overlapping blocks of size $B_1 \times B_1$ pixels, and store their descriptors in $\mathcal{B}_0$.
3. Extract a descriptor $(\mathbf{x}, B)$ from $\mathcal{B}_0$, i.e. $\mathcal{B}_0 = \mathcal{B}_0 \backslash \{(\mathbf{x}, B)\}$.
4. If $B == B_L$ then go to 6.
5. If $W_r(\mathbf{X}_{\mathbf{x}}^B) < \gamma S^2$ then divide $\mathbf{X}_{\mathbf{x}}^B$ in non-overlapping blocks of next smaller size, and insert their descriptors in $\mathcal{B}_0$. Go to 3.
6. Insert the descriptor $(\mathbf{x}, B)$ into $\mathcal{B}$, i.e. $\mathcal{B} = \mathcal{B} \bigcup \{(\mathbf{x}, B)\}$.
7. If $\mathcal{B}_0$ is not empty then go to 3.

Finally, we can summarize the proposed stabilization algorithm in the following steps:

1. Convert the input images in the irradiance domain in accordance to (2).
2. Estimate the additive noise variance in each input image g_k.
3. Determine the block division (i.e. the set $\mathcal{B}$) in accordance to previous algorithm acting only on the reference image g_r.
4. Restore each block in $\mathcal{B}$ in accordance to (5).
5. Convert the resulted irradiance estimate $\hat{f}$, back to the image domain $\hat{I}(\mathbf{x}) = \text{CRF}(\hat{f}(\mathbf{x})\Delta t)$, based on the desired exposure time Δt. Alternatively, in order to avoid saturation and hence to extend the dynamic range of the captured image, one can employ a tone mapping procedure (e.g. [9]) for converting the levels of the irradiance image estimate into the limited dynamic range of the system.

The extension of the proposed procedure to color or multichannel images is done by applying (5) to each image channel, with weight values calculated as follows:

$$w_k(\mathbf{x}, \mathbf{y}) = \left[\prod_{c=1}^{C} w_k^c(\mathbf{x}, \mathbf{y})\right]^{\frac{1}{C}}, \tag{9}$$

where C denotes the number of color channels, and $w_k^c(\mathbf{x}, \mathbf{y})$ is the weight function (6), calculated based only on the c-th color channel.

3. EXPERIMENTS

Comparisons between the proposed method and different other methods applied on public domain test images is shown in Table 1. In these experiments a single input frame was assumed. For the proposed method we used block sizes $B \in \{8, 4, 1\}$ with corresponding outer block sizes $U \in \{8, 4, 3\}$, and a search range $S = 9$. The method is compared against:

M1 the method proposed in our previous work [6],

M2 the Matlab's spatial local Wiener filtering,

M3 the hard thresholding of wavelet coefficients [10],

M4 the hard thresholding of curvelet coefficients [11].

Fig. 1, shows the performance achieved when using different numbers of input frames. For this experiment we used the "Lenna" image with an additive noise standard deviation of 15, and a search range $S = 3$ was set for the algorithm. The figure shows,with dashed line, the performance achieved by averaging the input frames after their global registration. In Fig. 1(a), it is assumed a perfect global registration between the input frames, and that the scene is static. A more practical scenario is simulated in Fig. 1(b), where some errors in the global alignment of the input images are inserted in the form of small perturbation of ±3 pixels translations, and ±1 degree rotations between input frames.

	Noise standard deviation			
	10	15	20	25
Lenna (512×512)				
Proposed method	**34.73**	**32.87**	**31.52**	**30.37**
(M1) The method in [6]	31.88	30.02	28.76	27.71
(M2) Wiener filtering	33.73	31.25	29.09	27.25
(M3) Wavelet de-noising	30.77	29.02	27.80	26.87
(M4) Curvelet de-noising	33.69	32.32	31.30	30.35
Barbara (512×512)				
Proposed method	**32.86**	**30.65**	**29.05**	**27.78**
(M1) The method in [6]	30.07	27.21	25.44	24.32
(M2) Wiener filtering	29.80	28.24	26.76	25.45
(M3) Wavelet de-noising	27.26	25.04	23.66	22.91
(M4) Curvelet de-noising	29.19	26.59	25.34	24.68
Cameraman (256×256)				
Proposed method	**33.48**	**31.09**	**29.61**	**28.36**
(M1) The method in [6]	31.36	28.99	27.27	25.90
(M2) Wiener filtering	30.90	29.40	27.85	26.41
(M3) Wavelet de-noising	28.24	26.06	24.57	23.60
(M4) Curvelet de-noising	29.53	27.58	26.40	25.66

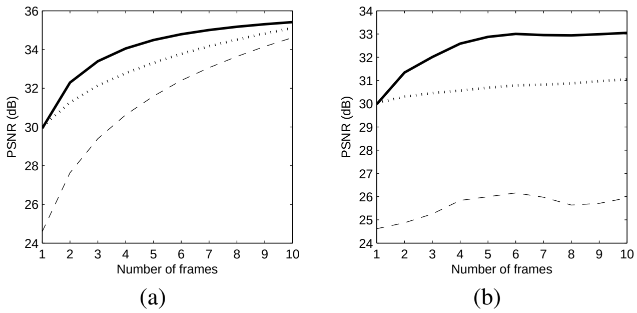
Table 1. PSNR results in decibels achieved with different approaches.

One example of low light imaging is shown in Fig. 2. In this experiment four input image frames have been captured in low light. During the time the individual frames have been captured the camera was slightly moving, and also various objects were moving in the scene, as reveal by Fig. 2 (b). Applying the proposed algorithm we can recover the latent scene image at any moment when an individual frame was captured. Fig. 2 (c) and (d) show two such examples based on using two different input frames as reference. The ability of the proposed algorithm to reduce the noise present in the individual input frames is exemplified by the detail shown in Fig. 2 (e) and (f).

The individual frames can be also degraded by blur, which ultimately may affect the final result. Fig. 3, shows such an example where some of the input frames are degraded (e.g. Fig. 3 b). If all four frames of this set would be registered and added together then the blur regions of various frames would degrade the quality of the final image as shown in Fig. 3 (c). The proposed method is able to reduce the effect of blur image regions in individual frames (Fig. 3 d), providing that the reference frame is selected based on a sharpness criteria.

4. CONCLUSIONS

In this paper we introduced an image stabilization approach based on fusing visually similar image blocks available in multiple observed images of the scene. The block sizes are automatically adapted to the image content such that to optimize the trade off between quality and complexity. The proposed method acts along the temporal (interframe), and spatial (intra-frame) dimensions, reducing the effect of camera and object motion. The experiments show that the proposed approach achieves high de-noising performance even with one input frame (Table 1), and that it improves with the number of input frames tolerating possible misalignment caused by camera motion (Fig. 1). The method has been demonstrated also on real image examples.



(a) (b)

Fig. 1. PSNR for different numbers of input frames fused in accordance to the proposed method (continuous line), using the method [6] (dotted line), or temporal filtering (dashed line). (a) shows the ideal case when there is no camera motion, and (b) shows the real case when there is camera motion.

5. REFERENCES

[1] Tony F. Chan and Chiu-Kwong Wong, "Total Variation Blind Deconvolution," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 370–375, 1998.

[2] Yu-Li You and M. Kaveh, "A regularization approach to joint blur identification and image restoration," *IEEE Trans. on Image Processing*, vol. 5, no. 3, pp. 416–428, Mar. 1996.

[3] Moshe Ben-Ezra and Shree K. Nayar, "Motion-Based Motion Deblurring," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 689–698, 2004.

[4] Marius Tico, Mejdi Trimeche, and Markku Vehvilainen, "Motion blur identification based on differently exposed images," in *Proc. of the IEEE International Conference of Image Processing (ICIP)*, Atlanta, GA, USA, Oct. 2006, pp. 2021–2024.

[5] Xinqiao Liu and Abbas El Gamal, "Synthesis of high dynamic range motion blur free image from multiple captures," *IEEE Transaction on Circuits and Systems-I*, vol. 50, no. 4, pp. 530–539, 2003.

[6] Marius Tico and Markku Vehvilainen, "Robust image fusion for image stabilization," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Honolulu, USA, 2007.

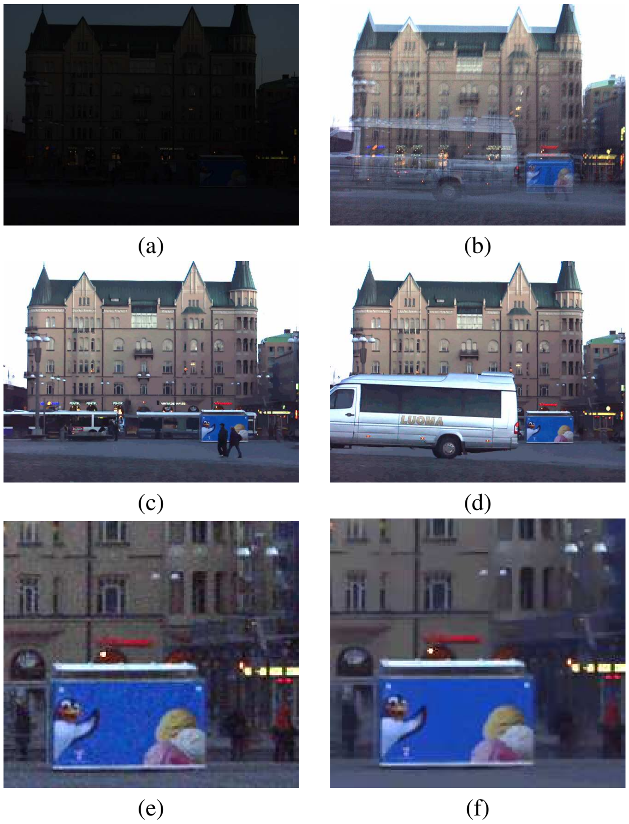
[7] Paul E. Debevec and Jitendra Malik, "Recovering High Dynamic Range Radiance Maps from Photographs," in *Proc. of International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 1997.

[8] Tomoo Mitsunaga and Shree K. Nayar, "Radiometric self calibration," in *Proc. of Conference on Computer Vision and Pattern Recognition*, 1999.

[9] Duan Jiang and Qiu Guoping, "Fast tone mapping for high dynamic range images," in *Proc. of 17th Intl. Conf. on Pattern Recognition (ICPR)*, 2004, vol. 2, pp. 847–850.

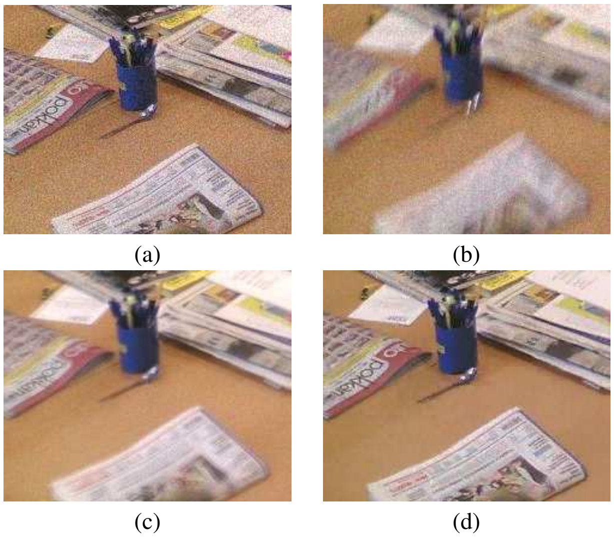
[10] D. L. Donoho and I. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, pp. 425–455, 1994.

[11] Jean-Luc Starck, Emmanuel J. Candes, and David L. Donoho, "The Curvelet Transform for Image Denoising," *IEEE Trans. on Image Processing*, vol. 11, no. 6, pp. 670–684, 2002.



(a) (b) (c) (d) (e) (f)

Fig. 2. Low light imaging example: (a) input short exposed image, (b) overlapped 4 input images, (c,d) two different results by the proposed algorithm showing the scene at different reference moments, (e,f) detail showing the noise reduction achieved by the proposed method (f) in comparison with a gain increased input image (e).



(a) (b) (c) (d)

Fig. 3. Dealing with degraded input frames: (a) fragment from the reference frame out of four input frames used by the algorithm, (b) one input frame heavily corrupted by blur, (c) the result obtained by averaging the registered frames, and (d) the result of our algorithm.