Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

DIPLOMOVÁ PRÁCE

Jozef Sabo

Odstranění rozmazání pomocí dvou snímků s různou délkou expozice

Katedra softwarového inženýrství

Vedoucí diplomové práce: Ing. Filip Šroubek, PhD.
Studijní program: Softwarové inženýrství

2010

Autor děkuje vedoucímu diplomové práce, Ing. Filipu Šroubkovi, PhD., za jeho trpělivost, ochotu, poskytnuté materiály a cenné rady při psaní diplomové práce a své rodině za vytrvalou podporu.

Prohlašuji, že jsem diplomovou práci napsal samostatně a výhradně s užitím citovaných pramenů. Souhlasím se zapůjčovaním práce a jejím zveřejňováním.

V Praze, dne 10.12.2010 Jozef Sabo

Název práce	Odstranění rozmazání pomocí dvou snímků s různou délkou expozice
Autor	Jozef Sabo
Katedra (ústav)	Katedra softwarového inženýrství
Vedoucí diplomové práce	Ing. Filip Šroubek, PhD.
E-mail vedoucího	sroubekf@utia.cas.cz
Klíčová slova	krátká, dlouhá, expozice, odstranění, rozmazání, šum, dekonvoluce

Abstrakt: V předložené práci studujeme metody odstranění rozmazání pomocí dvou snímků stejné předlohy s různou dobou expozicie, přičemž se soustřeďujeme na dvě hlavní kategorie těchto metod, tzv. dekonvoluční a nedekonvoluční. U obou kategorií rozebíráme jejich teoretické základy a zkoumáme jejich výhody a omezení. Samostatnou kapitolu věnujeme vyhodnocení a srovnání kategorií metod na testovacích datech (obrázky), k testování používáme metody implementovány v jazyku MATLAB. Účinnost zkoumaných metod srovnáváme i s vybraným odšumovacím algoritmem pracujícíms jedním vstupním obrázkem. Nesoustředíme se na výpočetní složitost srovnávaných algoritmů a pracujeme pouze s jednokanálovými obrázky.

Title	Image de-blurring using two images with different exposure times
Author	Jozef Sabo
Department	Department of software engineering
Supervisor	Filip Šroubek, PhD., UTIA
Supervisor's email	sroubekf@utia.cas.cz
Keywords	long, short, exposure, image, deblurring, denoising, deconvolution

Abstract: In the presented work we study the methods of image deblurring using two images of the same scene with different exposure times, focusing on two main approach categories, so called deconvolution and non-deconvolution methods. We present theoretical backgrounds on both categories and evaluate their limitations and advantages. We dedicate one section to compare both method categories on test data (images) for which we our MATLAB implementation of the methods. We also compare the effectiveness of said methods against the results of a selected single-image de-noising algorithm. We do not focus at computational efficiency of algorithms and work with single-channel images only.

1 Table of contents

2 Introduction

Image restoration[1] techniques have always been a subject of great interest within the domain of digital image processing. Their significance increases as digital photography undergoes rapid development with many digital imaging devices becoming readily available in countless forms, such as cell-phones, cameras and video cameras, to name a few.

We know from experience that it is quite difficult to obtain a quality image especially in insufficient lighting conditions requiring longer exposure times. In other words, taking "bad images" is easy. One of the most prevalent image degrading factors in photography is *motion blur*, caused either by motion in the photographed scene, the camera itself, or both. Suppression or complete removal of motion blur is highly desirable especially in hand-held devices.

According to (1), there are essentially two categories of approaches to deal with motion blur, the so-called *in-process* and *post-process*. The former focus at improving the conditions at which the image is being taken, usually by hardware means (image stabilizers, CMOS cameras), while the latter aim to correct the effects of motion blur after the image was taken. Widespread deployment of *in-process*-capable devices is however limited due to their high prices and as a result, the need for effective *post-process* algorithms arises.

In image processing, motion blur is modeled by convolution. If a *point-spread function* or *PSF* of motion blur is known (this applies to camera motion as local blurring caused by movement of objects in the scene tend to lead to *PSF* that is location-dependent - which is beyond the scope of this paper; however, some of the methods to be mentioned below are capable of handling space-variant blurring naturally) the original image can be recovered by a deconvolution algorithm, such as Lucy-Richardson (2). In most cases, there is little or no prior information on blur PSF which requires us to employ a *blind deconvolution* algorithm. First, a PSF is estimated from a given image or a set of images and subsequently, an existing *non-blind deconvolution* algorithm is used.

In case there is only one image available, results rarely prove satisfactory. This is given by the under-determined nature of the problem. We have to rely on generalized models of motion blur which are usually not capable of capturing complex PSF shapes. Unsatisfying estimates of blur PSF then lead to even more unsatisfying estimates of the original image due to iterative nature of most deconvolution algorithms.

Additional information obtained from multiple images of the same scene subject to varying degrees of degradation (blur, noise) can improve overall results. In this paper, we focus at the situation where we obtain two images of the same scene using different exposure times. The first image is taken using a long exposure time resulting in proper level of lighting but degraded by motion blur caused by camera shake, whereas the second image is taken using a short exposure time and is not

[1] Image restoration aims to reconstruct the original pre-degradation image as faithfully as possible as opposed to image enhancement, which aims to make desired features more visible

affected by motion blur, but is darker. Both images are affected by noise, which is directly proportional to the *ISO* sensitivity setting of the digital camera; we elaborate on this in greater detail in chapter 2.1.

There are several approaches how to restore the original image from the short and long exposure image pairs. These broadly fall in two categories – the *non-deconvolution* and *deconvolution* algorithms. The former do not perform deconvolution at any stage and try to utilize the image information in other ways as opposed to the latter, where deconvolution is performed at some point. Both categories feature methods with varying complexity, computational cost and efficiency, some of which will be described below.

The aim of this work is to analyze selected methods and evaluate their degree of suitability in various circumstances in terms of exposure times, signal-to-noise ratio improvements and other indicators based on experimental results. The methods are compared against themselves and a chosen single-image denoising algorithm to evaluate the benefits of additional image information in the form of improved quality of the restored image. We do not attempt to implement computationally effective algorithms at all costs nor do we provide an exhaustive list of all available methods. We abandon physical camera experiments in favor of simulated data to achieve the best degree of control over the experiment. Single-channel (grayscale) images are assumed.

2.1 ISO and image noise

In conventional photography, the photographic material's sensitivity to light is determined by the size of silver halide grains embedded in its emulsion. The larger a grain is, the more photons can it capture increasing the probability of exposure. Upon illumination, grains develop in an all-or-nothing fashion meaning that a grain decomposes into silver completely or not at all. This produces the characteristic "film grain" that is especially present in highly sensitive material used to shoot fast-moving scenes with very short shutter times.

Several systems were used to designate film sensitivity in the past. Among the most common were ASA, DIN and GOST. The ASA and DIN scales were incorporated into the new ISO standard published in 1987 (ISO 5800:1987) as the ISO arithmetic scale and logarithmic scale, respectively. The ISO logarithmic scale gradually fell out of use in favor of the ISO arithmetic scale.

The process of capturing an image in a digital camera is somewhat different. Photographic material is replaced by an electronic *image sensor* which consists of an array of individual cells capable of converting light into electrical signals. This signal is then amplified, digitized and stored. Known types of image sensors include the *CCD* (charge-coupled devices) or *CMOS* (complementary metal-oxide semiconductors). CCD cells store captured light as an electrical charge until they are read (one at a time) whereas circuitry attached to each CMOS cell converts light energy into voltage directly. Neither technology has a clear advantage; CMOS sensors are however cheaper to manufacture.

Digital cameras typically allow the user to select from several ISO settings. This is made possible by varying the amplification factor affecting the signal leaving the image sensor. Since no image sensor is completely free of noise, amplification of the signal also amplifies noise, resulting in a lowered signal-to-noise ratio. Foi in (3) demonstrated that it is possible to model digital camera noise as two independent Gaussian (accounting for electrical and thermal noise) and Poissonian (accounting for the photon capturing process) components as follows:

$$z(\boldsymbol{x}) = f(\boldsymbol{x}) + \eta_p(f(\boldsymbol{x})) + \eta_g(\boldsymbol{x}) \quad [1]$$

where $\boldsymbol{x}$ is the pixel coordinate, $f(\boldsymbol{x})$ is the original signal, $z(\boldsymbol{x})$ is the observed signal and η_p and η_g are Poissonian and Gaussian noise components, respectively. In terms of distributions, the equation becomes

$$\begin{aligned} \chi(f(\boldsymbol{x}) + n_p(f(\boldsymbol{x}))) &\sim \mathcal{P}(\chi f(\boldsymbol{x})) \\ n_g(\boldsymbol{x}) &\sim \mathcal{N}(0, b) \end{aligned} \quad [2]$$

where $\chi > 0$ and $b \geq 0$ are real scalar parameters and $\mathcal{N}$ and $\mathcal{P}$ denote normal (i.e. Gaussian) and Poissonian distributions, respectively. Using $E\{\}$ for the expected value and $var\{\}$ for the variance of a random variable, we obtain

$$E\{\chi(f(\boldsymbol{x}) + n_p(f(\boldsymbol{x})))\} = var\{\chi(f(\boldsymbol{x}) + n_p(f(\boldsymbol{x})))\} = \chi f(\boldsymbol{x}) \quad [3]$$

from the properties of the Poisson distribution. Since

$$E\{\chi(f(\boldsymbol{x}) + n_p(f(\boldsymbol{x})))\} = \chi f(\boldsymbol{x}) + \chi E\{n_p(f(\boldsymbol{x}))\} \quad [4]$$

and

$$\chi^2 var\{n_p(f(\boldsymbol{x}))\} = \chi f(\boldsymbol{x}) \quad [5]$$

it follows that

$$E\{n_p(f(\boldsymbol{x}))\} = 0 \text{ and } var\{n_p(f(\boldsymbol{x}))\} = f(\boldsymbol{x})/\chi \quad [6]$$

The Poissonian η_p thus has variable variance that depends on the value of $f(\boldsymbol{x})$, $var\{\eta_p(f(\boldsymbol{x}))\} = af(\boldsymbol{x})$ where $a = \chi^{-1}$. The Gaussian component η_g has constant variance equal to b. As a consequence, the total variance of the expression [1] can be expressed as

$$\sigma^2(f(\boldsymbol{x})) = af(\boldsymbol{x}) + b \quad [7]$$

Poissonian distribution can be approximated by normal distribution to a sufficient degree of accuracy

$$\mathcal{P}(\lambda) \approx \mathcal{N}(\lambda, \lambda) \quad [8]$$

which combined with [6] and [7] yields

$$\eta_p(f(\boldsymbol{x})) + \eta_g(\boldsymbol{x}) \sim \mathcal{N}(0, af(\boldsymbol{x}) + b) \qquad [9]$$

The relationship between the ISO setting of a particular camera and parameters a and b are explored in (4) and adopted for use in this paper to generate experimental data.

2.2 Model

For the purpose of deblurring algorithms, we present a mathematical model of the short and long exposure image pair as mentioned above. Let $f(\boldsymbol{x})$ be the image function of the original, discrete, non-degraded grayscale image N pixels in size. Let $g_2(\boldsymbol{x})$ be the image function of the original image subject to blurring and additive noise[2] and finally let $g_1(\boldsymbol{x})$ be the image function of the underexposed image subject to additive noise only. According to Tico (4) we have:

$$\begin{aligned} \alpha g_1(\boldsymbol{x}) &= f(\boldsymbol{x}) \qquad\qquad + n_1(\boldsymbol{x}) \\ g_2(\boldsymbol{x}) &= f(\boldsymbol{x}) \star d(\boldsymbol{x}) + n_2(\boldsymbol{x}) \end{aligned} \qquad [10]$$

where $\boldsymbol{x} = (x, y)$ are pixel coordinates, $d(\boldsymbol{x})$ is the point-spread function describing motion blur, α is the change in brightness as a result of shorter exposure (clearly $\alpha \leq 1$), $\star$ denotes the convolution operation and n_1, n_2 are noise terms. We assume additive Gaussian noise with $\mu_1 = \mu_2 = 0$ and $\sigma_1^2 \gg \sigma_2^2$ (where μ_i and σ_i^2 are the mean and variance respectively).

[2] Noise is present in every imaging device

3 Non-deconvolution methods

In this category we include methods that do not use deconvolution or minimization of an objective function to achieve the deblurring goal. Due to the fact that the experimental setup can be modified to produce the blurred and noisy image pair of nearly the same overall brightness (the ratio of shutter times would have be equal to the inverse ratio of ISO settings) we deem it unnecessary to explore the image statistics-based fusion such as the work of Razligh (5).

3.1 Tico's method (2009)

Tico (6) presents a relatively simple wavelet-based approach to blurred and noisy image fusion. The images are first decomposed into their respective wavelet coefficients. Then, multi-level coefficient blending is performed. Finally, inverse wavelet transform is performed yielding the result image.

We observe that the absolute difference between the blurred and noisy images is due to presence of noise in the short-exposed image and blurring in the other. We therefore aim for an estimator that emphasizes the short-exposed image where the absolute difference between the two images is larger and the long-exposed image otherwise. To achieve better separation between the signal and noise an image estimator is derived in the wavelet domain. The edge locations (i.e., large values in the difference signal), are emphasized at some scales whereas the noise variance is evenly distributed across the scale space. Considering an orthonormal wavelet transform of the two images, denoting by $G_i(k)$, the k-th wavelet transform and assuming the same overall brightness of both images, we have

$$\begin{aligned} G_1(k) &= F(k) + N_1(k) \\ G_2(k) &= F_b(k) + N_2(k) \end{aligned} \qquad [11]$$

Where F_b denotes the blurred image as a whole since the nature of blurring is not important in this case. Using the observation $\sigma_1^2 \gg \sigma_2^2$ we neglect the noisy coefficients $N_1(k)$ and the term $N_2(k)$ becomes $N(k)$.

We can now fuse the images together using different weights at different scales. Taking advantage of the de-correlation in the wavelet domain, we propose a minimum mean square error diagonal estimator of the original image in the form of a linear combination between the wavelet coefficients of the two images

$$\hat{F}(k) = G_1(k) + W(k)D(k) \qquad [12]$$

where $\widehat{F}(k)$ stands for the wavelet coefficients of the restored image, $D(k) = G_2(k) - G_1(k)$ denotes the difference signal between the wavelet coefficients of the two observed images, and $W(k)$ are weight coefficients. We can estimate the best weight $W(k)$ for each wavelet coefficient by minimizing the mean squared error

$$E[\|\hat{F}(k) - F(k)\|]_2^2 = E[\|G_1(k) - F(k) + W(k)D(k)\|]_2^2 \qquad [13]$$

whose derivative with respect to $W(k)$ equated with zero yields

$$W(k)E[\|D(k)\|^2] = E[D(k)N(k)] \qquad [14]$$

The computation of the weight $W(k)$ requires an estimate of the noise variance in the short-exposed image, and an estimate of the term $E[\|D(k)\|^2]$. In order to estimate noise variance in the short-exposed image the approach presented by Mallat in (7) where noise variance is calculated from the median of the finest-scale wavelet coefficients M_x as $\bar{\sigma} \approx M_x/0.6745$ is used. Given the fact that in practice noise is spatially variant over the image, we apply the wavelet-based noise estimate in the pixel neighborhood (e.g. 7×7). Finally, the $E[\|D(k)\|^2]$ is approximated with $\max(\sigma^2(k), avg(|D(k)|^2))$ where avg denotes local spatial average and $\sigma^2(k)$ is the noise variance at the spatial location that corresponds to the k-th wavelet coefficient.

As a consequence, the weight $W(k)$ emphasizes the short exposed image in areas of image transitions (edges etc.) whereas the blurred image is emphasized in smooth regions.

4 Deconvolution methods

In contrast to section 3 this category contains methods which first try to estimate the blur PSF from the blurred and noisy images and then perform deconvolution on the blurred image. We also include methods that use minimization of an objective function to achieve the same task.

4.1 Tico-Vehvilainen method (2006)

4.1.1 Point-spread function estimation

The method of PSF estimation, described in (4) is built upon the model described above and tries to express the PSF in terms of posterior probability. Based on the known images g_1 and g_2 can we write

$$p(d|g_1, g_2) = \frac{p(g_2|g_1, d)p(d)p(g_1)}{p(g_1, g_2)} \qquad [15]$$

where, retaining the terms that depend on d, the objective function to be minimized by the maximum a posteriori estimate of the PSF can be written as

$$J(d, \alpha) = -\log\ p(g_2|g_1, d) - \ \log\ p(d) \qquad [16]$$

As a consequence of the model which assumes Gaussian noises of variances $\sigma_1^2 \gg \sigma_2^2$, the conditional probability density function $p(g_2|g_1, d)$ is a multivariate Gaussian with mean $\alpha g_1 \star d$ and a non-diagonal covariance matrix. For tractability of the solution, only the diagonal elements of the covariance matrix are considered which are given by

$$\sigma^2(d) = \sigma_2^2 + \sigma_1^2 \sum_{x \in \Psi} d(\boldsymbol{x})^2 \qquad [17]$$

from which we obtain the following simplified model of the conditional probability density function

$$-\log p(g_2|g_1, d) \sim \frac{1}{2\sigma^2(d)} \sum_{\boldsymbol{x} \in \Omega} n(\boldsymbol{x})^2 + \frac{N}{2} \log \sigma^2(d) \qquad [18]$$

where N is the number of image pixels and

$$n(\boldsymbol{x}) = g_2(\boldsymbol{x}) - \ \alpha g_1(\boldsymbol{x}) \star d(\boldsymbol{x}) \qquad [19]$$

The second term in the equation [15] describes the model of the blur PSF. If we assume that the camera only undergoes translational motion during the exposure time, we may consider the PSF to be space invariant and regard it as a projection of the camera's spatial motion onto the image plane, where it assumes the typical curved "ridge" shape. This ridge appearance is imposed on the PSF by defining the prior probability density function as

$$-\log p(d) \sim \frac{\lambda}{2} \sum_{\boldsymbol{x} \in \Psi} [1 - m(\boldsymbol{x})] d(\boldsymbol{x})^2 \qquad [20]$$

where $m(\boldsymbol{x})$ denotes the *indicator function*, which equals 1 if $\boldsymbol{x}$ belongs to the PSF ridge and 0 otherwise.

As a result of the physical constraints on the camera motion speed and acceleration, the PSF ridge can be assumed continuous and differentiable. Consequently, in most of its points $\boldsymbol{x}$, the direction $\theta(\boldsymbol{x})$ tangent to the ridge path is well defined. Based on this observation and aiming for the ridge-like shape of the blur PSF, we define the *ridge function* as equal to 1 if $\mathrm{d}(\boldsymbol{x}) \geq d(\boldsymbol{y})$ for any $y \in N(\boldsymbol{x})$ and 0 otherwise. $N(\boldsymbol{x})$ denotes the local neighborhood of the point $\boldsymbol{x}$ selected in the direction of $\theta(\boldsymbol{x})$ which is orthogonal to the local ridge orientation.

In practice, it is not possible to calculate m directly, since we do not know the blur PSF. We can apply the same approach onto on an intermediate estimate of the blur PSF, where the local ridge orientation at each $\boldsymbol{x} \in \Psi$ is calculated by the texture orientation estimator from (8).

Joining [18] and [20] we obtain the final form of the objective function. This can be minimized by the gradient descent algorithm imposing the constraints $\sum_{\boldsymbol{x} \in \Psi} d(\boldsymbol{x}) = 1$ a $\mathrm{d}(\boldsymbol{x}) \geq 0, \boldsymbol{x} \in \Psi$.

The gradient of the objective function is given as

$$
\begin{aligned}
J_d(d,\alpha) &= \frac{\partial J(d,\alpha)}{\partial d} \\
&= \frac{-\alpha g_1(-\boldsymbol{x}) \star n(\boldsymbol{x})}{\sigma^2(d)} \\
&+ \frac{d(\boldsymbol{x})\sigma_1^2}{\sigma^2(d)}\left[N - \frac{\sum_{x\in\Omega} n(\boldsymbol{x})^2}{\sigma^2(d)}\right] \\
&+ \lambda[1 - m(\boldsymbol{x})]d(\boldsymbol{x})
\end{aligned} \qquad [21]
$$

and

$$
J_\alpha(d,\alpha) = \frac{\partial J(d,\alpha)}{\partial \alpha} = \frac{\sum_{x\in\Omega}[g_1(\boldsymbol{x}) \star d(\boldsymbol{x})]n(\boldsymbol{x})}{\sigma^2(d)} \qquad [22]
$$

where $\Omega \subset R^2$ denotes the image support. The parameter α is estimated at each iteration by equating the previous equation with zero:

$$
\alpha = \frac{\sum_{x\in\Omega} g_2(\boldsymbol{x})[g_1(\boldsymbol{x}) \star d(\boldsymbol{x})]}{\sum_{x\in\Omega}[g_1(\boldsymbol{x}) \star d(\boldsymbol{x})]^2} \qquad [23]
$$

The initial α estimate can be obtained as the ratio of the means of the two images:

$$
\alpha_0 = \sum_{x\in\Omega} g_2(\boldsymbol{x}) \Big/ \sum_{x\in\Omega} g_1(\boldsymbol{x}) \qquad [24]
$$

Note that the prior term in [23] is strongly dependent on the current estimate of d. Therefore the initial value of λ is set to zero and then to a high value after several iterations in order to force ridge-like appearance on the blur PSF.

The iterative minimization algorithm could start from an arbitrary initial guess of the blur PSF. However, in order to speed up the process an initial value can be used, which can be obtained by the following algorithm.

4.1.1.1 Initial PSF estimate

Based on the model we can write

$$
g_2(\boldsymbol{x}) = \alpha g_1(\boldsymbol{x}) \star d(\boldsymbol{x}) + n(\boldsymbol{x}) \qquad [25]
$$

where $n(\boldsymbol{x}) = n_2(\boldsymbol{x}) - n_1(\boldsymbol{x}) \star d(\boldsymbol{x})$. Neglecting the non-diagonal terms of the covariance matrix of $n(\boldsymbol{x})$ and by using the property $\sigma_2^2 \ll \sigma_1^2$ the Wiener filter estimation of the blur PSF

$$
D(\omega) = \frac{\alpha G_1^*(\omega) G_2(\omega)}{\alpha^2 |G_1(\omega)|^2 + \sigma_1^2} \qquad [26]
$$

where capital letters denote the Fourier transforms of their respective signals and α originates in [10]. The initial blur PSF is then obtained by the inverse Fourier transform of $D(\omega)$. A practical implementation is described by the following algorithm.

Input: Two images g_1, g_2 and an approximate estimate of the PSF support size, i.e. $S_1 \times S_2$.

Output: The initial PSF estimate.

Algorithm:

- Select several blocks of size $W_1 \times W_2$ ($W_1 > S_1$ and $W_2 > S_2$) from the blurred image g_2. Selection is based on the standard deviation of each block; blocks of higher standard deviation are preferred due to greater probability of them containing significant details or transitions. Blocks are labeled g_2^k for $k = 1..K$ where K is the amount of selected blocks. The corresponding blocks from the image g_1 are labeled g_1^k.
- We average K estimates given by equation [26] from corresponding blocks g_1^k, g_2^k . Fourier transforms are calculated by FFT.
- The final PSF estimate is obtained by selecting the central part ($S_1 \times S_2$) from the inverse $W_1 \times W_2$ Fourier transform of the average computations in the previous point.

It is however possible to use the whole images g_1, g_2 as the input of equation [26] at the cost of lower quality and slower algorithm convergence.

4.1.2 Blur removal

Estimated PSF is used to remove blur by using the Lucy-Richardson algorithm, which is implemented as function *deconvlucy* in MATLAB.

4.2 Tico-Vehvilainen method (2007)

An "upgrade" of the method introduced in chapter 4.1 was published in 2007 by Tico and Vehvilainen. Our base model is extended by an additional term $e(\boldsymbol{x})$

$$\begin{aligned} \alpha e(\boldsymbol{x}) \star g_1(\boldsymbol{x}) &= f(\boldsymbol{x}) \qquad\quad + n_1(\boldsymbol{x}) \\ g_2(\boldsymbol{x}) &= f(\boldsymbol{x}) \star d(x) + n_2(\boldsymbol{x}) \end{aligned} \qquad [27]$$

that denotes the blur PSF of the underexposed image which (as image is not subject to motion blur) is identical to the Dirac function $\delta(\boldsymbol{x})$. The additional PSF was introduced to model the residual blurring between $g_1(\boldsymbol{x})$ and intermediate estimates of the original image $f(\boldsymbol{x})$ in the algorithm to be described below. The n_1 and n_2 are Gaussian noise of zero mean and variances σ_1^2 and σ_2^2 that satisfy $\sigma_2^2 \ll \sigma_1^2$.

The joint posterior probability density function of the original image $f(x)$ and both PSFs can be expressed as

$$p(f, d, e|g_2, g_1) \sim p(g_2|f, d)p(g_1|f, e)p(f)p(d, e) \qquad [28]$$

from where leaving out the terms that do not depend on f, d or e we obtain the maximum a posteriori (MAP) objective function

$$Q(f, d, e) = -\log p(g_2|f, d) - \log p(g_1|f, e) - \log p(f) - \log p(d, e) \qquad [29]$$

The first two terms of the previous equation can be derived from the model [10]

$$\begin{aligned} -\log p(g_2|f, d) &\sim (\lambda_2/2) \sum_{\boldsymbol{x}\in\Omega} [g_2(\boldsymbol{x}) - f(\boldsymbol{x}) \star d(\boldsymbol{x})]^2 \\ -\log p(g_1|f, e) &\sim (\lambda_1/2) \sum_{\boldsymbol{x}\in\Omega} [\alpha g_1(\boldsymbol{x}) \star e(\boldsymbol{x}) - f(\boldsymbol{x})]^2 \end{aligned} \qquad [30]$$

where $\lambda_i = \sigma_i^{-2}$ for $i = 1{,}2$ and Ω denotes the PSF support. To avoid over-smoothing the image we adopt a discrete form of the total variation (TV) prior

$$-\log p(f) \sim \gamma \sum_{x\in\Omega} |\nabla f(x)| \qquad [31]$$

where ∇ denotes the spatial gradient operator a γ is the prior weight balances the confidence between the prior and observations. The gamma distribution is assumed, i.e. $\gamma \sim \Gamma(\gamma|a, b)$. The parameter γ is updated at each iteration based on the current estimate of the original image f.

The model used for both PSFs was chosen by the authors to optimized when e becomes identical to Dirac delta function, that is

$$-\log p(d, e) \sim \frac{\beta_1}{2} \sum_{\boldsymbol{x}\in\Omega} [d(\boldsymbol{x}) \star e(\boldsymbol{x}) - d(\boldsymbol{x})]^2 + \frac{\beta_2}{2} \sum_{\boldsymbol{x}\in\Omega} [e(\boldsymbol{x}) - \delta(\boldsymbol{x})]^2 \qquad [32]$$

where β_1 and β_2 are positive values weighing the importance of the PSF prior. Joining the previous three equations we obtain the final form of the objective function whose optimization is achieved by the following algorithm.

Input: Images g_1, g_2 and an estimate of the size of the PSF support.
Output: The image f.
Initialization: $k = 0, f^k = g_1, \epsilon_k = \infty$.
Iteration:

1. Minimize with fixed f: $\hat{d}, \hat{e} \leftarrow argmin_{d,e} Q(f^{(k)}, d, e)$
2. Update PSF: $d^{(k)}(\boldsymbol{x}) = \hat{d}(\boldsymbol{x}) \star \hat{e}(\boldsymbol{x})$
3. Enhance d^k by removing noise and normalization
4. Minimize with fixed d: $f^{(k+1)} \leftarrow argmin_f Q(f, d^{(k)}, \delta)$ by iterating from the initial estimate $f = g_2$
5. Calculate $\epsilon_{k+1} = \|f^{(k+1)} \star d - g_1\|$
6. If $\epsilon_{k+1} > \epsilon_k$, return $f = f^{(k)}$

In the first step of the algorithm the objective function is minimized with respect to d and e. This can be achieved by solving the system of equations obtained when equating with zero the gradients of Q with respect to d and e.

The third step of the algorithm aims to improve the representation of d^k by canceling its noisy coefficients. To distinguish the real PSF coefficients from the noisy ones the PSF signal is analyzed at multiple levels of smoothness obtained by iterative low-pass filtering. A threshold based on standard deviation at a given level is established and all coefficients below that threshold are cancelled. Finally, we cancel in d^k all coefficients that have been cancelled at any of the levels. The remaining coefficients in d^k are then normalized to sum up to 1, i.e. $\sum d(\boldsymbol{x}) = 1$.

The fourth step of the algorithm minimizes the objective function with respect to f for the given PSF. The gradient of the objective function with respect to f yields

$$\begin{aligned}\nabla_f Q = \lambda_2[f(\boldsymbol{x}) \star d^{(k)}(\boldsymbol{x}) - f(\boldsymbol{x})] \star d^{(k)}(-\boldsymbol{x}) \\ +\lambda_1[f(\boldsymbol{x}) - \alpha g_1(\boldsymbol{x})] + \gamma\nabla[w(\boldsymbol{x})\nabla f(\boldsymbol{x})]\end{aligned} \quad [33]$$

where $w(\boldsymbol{x}) = \frac{1}{|\nabla f(\boldsymbol{x})|}$ is the diffusive coefficient. The objective function is minimized by conjugate gradient method while the diffusive coefficient is lagged one iteration behind. Convergence is relatively fast due to conjugate gradient iteration properties and is stopped when the relative change in the objective function between two iterations becomes less than a given threshold. It is important to stress that the γ parameter is updated in each iteration based on the current estimate of the image f and at iteration i we have

$$\gamma_i = \frac{a-1}{b + \left(\frac{1}{N}\right)\sum_{\boldsymbol{x}\in\Omega}|\nabla f_i(\boldsymbol{x})|} \quad [34]$$

where N denotes the number of pixels in the image and a and b are parameters of the Γ-distribution imposed on the weight γ.

5 Results

5.1 Experimental setup

5.2 Evaluation

6 Conclusion

6.1 Possible future improvements

7 Bibliography

1. **Jia J., Sun J., Tang C.K., Shum H.Y.** *Bayesian correction of image intensity with spatial consideration.* Computer Science Department, Hong Kong University of Science and Technology, 2007.
2. **Richardson, W. H.** *Bayesian-Based Iterative Method of Image Restoration.* JOSA, 1972. Vol. 62.
3. **Foi A., Trimeche M., Katkovnik V., Egiazarian K.** *Practical Poissonian-Gaussian noise modeling and fitting for single-image raw-data.* IEEE Transactions, 2007.
4. **Tico M., Vehvilainen M.** *Estimation of motion blur point spread function from differently exposed images.* Nokia Research Center, Tampere, Finland, 2006.
5. **Razligh Q.R., Kehtarnavaz N.** *Image blur reduction for cell-phone cameras via adaptive tonal correction.* Department of Electrical Engineering, University of Texas at Dallas, 2007.
6. **Tico M., Pulli K.** *Image enhancement via blur and noisy image fusion.* Nokia Research Center, Palo Alto, CA, USA, 2009.
7. **Mallat, S.** *A wavelet tour of signal processing, the sparse way.* Elsevier, Burlington, MA, 2009.
8. **Rao, A.R.** *A Taxonomy for Texture Description and Identification.* Springer Verlag, 1990.
9. **Liu X., Gamal A.** *Simultaneous image formation and motion blur restoration via multiple capture.* Proc. Int. Conf. Acoustics, Speech, Signal Processing, 2001.
10. **Yuan L., Sun J., Quan L., Shum H.Y.** *Image Deblurring with blurred/noisy image pairs.* Computer Science Department, Hong Kong University of Science and Technology, 2007.
11. **Tico M., Vehvilainen M.** *Image stabilization based on fusing the visual information in differently exposed images.* Nokia Research Center, Tampere, Finland, 2007.
12. **Beneš B., Felkel P., Sochor J., Žára J.** *Moderní počítačová grafika, 2. vydání.* Computer Press, 2004.
13. **Biedl T., Chan T., Demaine E., Fleischer R., Golin M., King J., IanMun J.** *Fun Sort - or the Chaos of Unordered Binary Search.* Discreate Applied Mathematics, vol. 144, p. 231-236, 2004.
14. **Gooch M., Reinhard E., Ashikmin E., Shirley P.** *Color transfer between images.* IEEE Computer Graphics and Applications, p. 34-40, 2001.
15. **Petteri, Ojala M.** *Dependence of the parameters of digital image noise model on ISO number, temperature and shutter time.* s.l. : Tampere University of Technology, Department of Automation Science and Engineering, Tampere, Finland, 2008.