IMAGE PROCESSING ON LINE

Published in Image Processing On Line on 2013–10–23.
Submitted on 2013–02–26, accepted on 2013–07–02.
ISSN 2105–1232 © 2013 IPOL & the authors CC–BY–NC–SA
This article is available online with supplementary materials,
software, datasets and online demo at
https://doi.org/10.5201/ipol.2013.77

Recovering the Subpixel PSF from Two Photographs at Different Distances

Mauricio Delbracio[1], Andrés Almansa[2], Pablo Musé[3]

[1] CMLA, ENS Cachan, France (mdelbra@fing.edu.uy)
[2] TELECOM ParisTech, France (andres.almansa@telecom-paristech.fr)
[3] IIE, UdelaR, Uruguay (pmusel@fing.edu.uy)

Communicated by Pascal Monasse

This IPOL article is related to a companion publication in the SIAM Journal on Imaging Sciences:
M. Delbracio, A. Almansa, J.M. Morel, and P. Musé. "Subpixel Point Spread Function Estimation from Two Photographs at Different Distances". SIAM Journal on Imaging Sciences 5(4):1234–1260, 2012.
http://dx.doi.org/10.1137/110848335

Abstract

In most typical digital cameras, even high-end digital single lens reflex ones (DSLR), the acquired images are sampled at rates below the Nyquist critical rate, causing aliasing effects. In this work we describe a new algorithm for the estimation of the point spread function (PSF) of a digital camera from aliased photographs, that achieves subpixel accuracy. The procedure is based on taking two parallel photographs of the same scene, from different distances leading to different geometric zooms, and then estimating the kernel blur between them.

Source Code

ANSI C source code to produce the same results as the demo is accessible at the IPOL web page of this article[1]. Future software releases and updates will be posted at https://github.com/mdelbra/two-photos-psf-estim.

Keywords: psf, mtf, blur, calibration, aliasing, subpixel

Disclaimer

This work publishes only the PSF estimation algorithm as described below. It is not the object of this work to study the problem of image registration. For this purpose the demo code uses the ORSA–homography routines [4], which are based on SIFT keypoints. All these subroutines may be eventually updated or replaced by other image registration subroutines.

[1] https://doi.org/10.5201/ipol.2013.77

Mauricio Delbracio, Andrés Almansa, Pablo Musé, *Recovering the Subpixel PSF from Two Photographs at Different Distances*, Image Processing On Line, 3 (2013), pp. 242–251. https://doi.org/10.5201/ipol.2013.77

1 Introduction

In most digital cameras, and even in high-end DSLR, the acquired images are sampled at rates far below the Nyquist critical rate, causing aliasing effects. In this work we introduce a blind algorithm for the subpixel estimation of the point spread function of a digital camera from aliased photographs. The numerical procedure simply uses two fronto-parallel photographs of any planar textured scene at different distances. The mathematical theory developed by Delbracio et al. [1] proves that the camera PSF can be derived from the relative kernel between the two images. Experimental evidence shows the well-posedness of the problem and the convergence of the proposed algorithm to the camera in-focus PSF. An experimental comparison of the resulting PSF estimates shows that the proposed algorithm reaches the accuracy levels of the best non-blind state-of-the-art methods.

Light diffraction, lens aberrations, sensor averaging and antialiasing filters are some of the inherent camera factors that unavoidably introduce blur in images. The blur that results from the combination of all these factors can be modeled locally as a convolution kernel known as *Point Spread Function* (PSF), that corresponds to the space variant impulse response of the optical system.

PSF estimation procedures can be either blind or non-blind. Non-blind approaches assume perfect knowledge of a specially designed calibration pattern, and perform local kernel estimation by comparing one or several photographs of the calibration pattern with the ideal calibration pattern. Blind approaches estimate the PSF from a single or a set of photographs from one or several scenes, whose exact knowledge or exhaustive descriptions are not required. They do assume, however, that the scenes involved in the estimation follow some statistical model of sharp images, or include a significant amount of geometric cues such as sharp edges. Most of the blind PSF approaches attempt to detect edges, which are modeled as pure step-edge functions convolved with the PSF kernel. In this setting, blind estimation is very ill-posed; to solve the inverse problem, the solution space has to be constrained by considering kernels with a parametric model or with strong regularity assumptions. Blind estimation techniques are in general much less accurate than their non-blind counterparts. The proposed formulation ensures regularization-free subpixel recovery of the PSF, from a pair of photographs of the same scene. The pictures have to be acquired at sufficiently different distances, with fixed camera configuration. In this sense, the proposed PSF calibration procedure is as simple as for any blind calibration method.

This article is organized as follows. Section 2 describes the general mathematical digital camera model used for PSF estimation. In section 3 we outline the mathematical analysis that motivates the proposed PSF estimation algorithm. The complete analysis of the theory underlying the proposed PSF estimation algorithm is presented in the companion paper by Delbracio et al. [1]. In section 4 a detailed description of the implemented algorithm is given while in section 5 experimental results generated with real camera data are presented. In appendix 7 we summarize some tips concerning the set-up for an accurate estimation. ANSI C source code to produce the same results as the demo is accessible on the article web page https://doi.org/10.5201/ipol.2013.77.

2 The Camera Model

An accurate estimation of the PSF requires a proper modeling of the digital image formation process. The simple pinhole camera model commonly used in computer vision only tells how to obtain a continuous planar scene w from the 3D world. When the 3D world is composed of a single planar scene u, the latter will be distorted at the image plane to $w = u \circ D$ by a homography D.

The distortion D may take the form of a more general (but regular) diffeomorphism, when the scene is a regular close-to-planar surface, or when the geometric distortion due to the optical system is taken into account.

For the purpose of PSF estimation this simple model needs to be augmented with at least:

1. a modeling of continuous to digital conversion at the image plane, *i.e.* a *sampling* operator $\mathbf{S}_1$ and additive *noise* $\mathbf{n}$ due to measurement uncertainties, and
2. the *blurring* kernel h due to intrinsic camera characteristics, such as diffraction when light goes through a finite aperture, light averaging within the sensor, lens aberrations, etc. Note that we do not consider here blur effects (like motion or defocus blur) that may change from one snapshot to another. This means that particular attention has to be paid while capturing images for PSF estimation, in order to minimize those variable blur sources.

The whole image formation process can be summarized in a single equation

$$\mathbf{v} = g(\mathbf{S}_1\left((u \circ D) * h\right) + \mathbf{n},$$

where $g(\cdot)$ is a monotone non-decreasing function that describes the non-linear sensor response. If the camera is working outside of the saturation zone, the response should be linear, since we are dealing directly with RAW images. Therefore, we will consider that the function g is linear so we can include its effect inside the image u. Hence, in the sequel, the image formation model will be

$$\mathbf{v} = \mathbf{S}_1\left((u \circ D) * h\right) + \mathbf{n}. \tag{1}$$

Suppose that the PSF h is s-band-limited, that is $\operatorname{supp}(\hat{h}) = [-s\pi, s\pi]^2$. Then, if sampled at least at a rate s, the Nyquist sampling theorem guarantees perfect reconstruction of h from its samples. We are actually interested in the case $s > 1$, usual for digital cameras.

Notation In the sequel, $\hat{f}$ denotes the Fourier transform of a function f. We denote by $\mathbf{I}_1$ the Shannon-Whittaker interpolator defined as $\mathbf{I}_1\mathbf{u}(\mathbf{x}) = \sum_{\mathbf{k}} \mathbf{u}(\mathbf{k})\operatorname{sinc}(\mathbf{x}-\mathbf{k})$. Let us denote by $\mathcal{S}_s$ the s-to-1-resampling operator $\mathcal{S}_s = \mathbf{S}_1 H_s \mathbf{I}_1$ and the continuous homothecy $H_\lambda u(x,y) = \lambda^2 u(\lambda x, \lambda y)$. (*i.e.* $\lambda > 1$ is a zoom-out). The digital Nyquist homothecy operator of parameter α is defined by $\mathbf{H}_\alpha \mathbf{u} := \mathbf{S}_1 W_1 H_\alpha \mathbf{I}_1 \mathbf{u}$, where W_1 is the ideal low pass filter at frequency π. We also denote the linear map associated to the convolution with a digital image $\mathbf{u}$ by $\mathbf{C}[\mathbf{u}]$. Let $\mathbf{L}$ be a linear operator, and denote by $\mathbf{L}^*$ its adjoint, and by $\mathbf{L}^+$ its left pseudo-inverse $\mathbf{L}^+ := (\mathbf{L}^*\mathbf{L})^{-1}\mathbf{L}^*$ (when it exists).

3 PSF Estimation from Two Unknown Scaled Images

Assume we perfectly know the latent sharp image u that produced the blurry aliased observation $\mathbf{v}$. Under this non-blind assumption solving for the PSF amounts to solve an inverse problem governed by the image formation model (1). The common approach that relies on regularization techniques allows to correctly recover the geometric and radiometric distortions (D and g), but severely distorts the high-frequency components of h. Notwithstanding, this non-blind inverse problem is well posed as long as a white noise image u is chosen as the calibration pattern [2], thus paving the way for unbiased estimation of h (because no regularization is used) up to unprecedented levels of accuracy. On the contrary, the widespread calibration patterns based on step-edges (see for instance the paper of Joshi et al. [3]) lead to ill-posed inverse problems, thus precluding accurate estimation of h at subpixel rate unless the family of admissible kernels h is drastically reduced by regularization or other techniques.

Between these two extremes many highly textured natural scenes exist which, while not being optimal, still lead to a well-posed inverse problem. Such images are exploited in this article in order to circumvent the non-blind hypothesis, by taking two snapshots of the same scene. More precisely, in the following, we show that recovery of camera PSF is possible from the estimate of the kernel blur between images of two different views. The presentation is divided in two parts. First we characterize this kernel for a pair of fronto-parallel views of a planar scene, and we give precise conditions under which this inter-image kernel can be estimated. Then we show that the camera PSF can be derived from the inter-image kernel, under very weak and reasonable conditions.

3.1 Estimating the Relative Blur between Two Images

Suppose we have two digital images $\tilde{\mathbf{v}}_1, \tilde{\mathbf{v}}_2$ of the same planar scene u, captured from different distances in a fronto-parallel position, with negligible rotation around the optical axis. Let λ_1, λ_2 denote the corresponding zooms between the scene and each of the images. This can be written

$$\tilde{\mathbf{v}}_i = \mathbf{S}_1 H_{\lambda_i} u * h + \mathbf{n}_i = \mathbf{S}_1 v_i + \mathbf{n}_i = \mathbf{v}_i + \mathbf{n}_i \quad \text{for } i = 1, 2,$$

where we have called $v_i = H_{\lambda_i} u * h$ and $\mathbf{v}_i = \mathbf{S}_1 v_i$. We assume that the acquisition distances are such that $s\lambda_1 < \lambda_2$, the importance of this will be clear later.

Let v_1, v_2 be two fronto-parallel continuous views of the same scene, acquired from different distances $\lambda_1 < \lambda_2$ respectively. We call *inter-image kernel* between v_1 and v_2, any kernel k satisfying

$$v_2 = H_{\frac{\lambda_2}{\lambda_1}} v_1 * k.$$

If we set $\lambda = \frac{\lambda_2}{\lambda_1}$, it can be shown [1] that the inter image kernel satisfies

$$H_\lambda h * k = h. \tag{2}$$

If $\hat{u}$ does not vanish inside $[-s\frac{\pi}{\lambda_2}, s\frac{\pi}{\lambda_2}]^2$ then the inter image-kernel is unique and only depends on h and λ. One simple way of estimating k is by considering the least squares estimator

$$\mathbf{k}_e := \left(\mathcal{S}_s \mathbf{C}[\mathbf{H}_{\frac{\lambda}{s}} \tilde{\mathbf{v}}_1]\right)^+ \tilde{\mathbf{v}}_2.$$

This estimator appears naturally from the definition of the inter-image kernel $v_2 = H_\lambda v_1 * k$, and by using $\tilde{\mathbf{v}}_1$ and $\tilde{\mathbf{v}}_2$ as approximations of $\mathbf{v}_1$ (the closest high resolution image) and $\mathbf{v}_2$ (the farthest low resolution image) respectively. The estimation is performed on a discrete grid of $s\times$ the camera sensor resolution. The resulting estimation will be accurate as long as the noise is much smaller than the signal power and the image $\mathbf{v}_1$ is not very aliased. See the article of Delbracio et al. [1] for more quantitative results and mathematical details.

3.2 Recovering the Camera PSF

Notice that h appears on both sides of (2). However, it can be obtained from k by calculating the following limit

$$h = \lim_{n\to\infty} H_{\lambda^n} k * \ldots * H_\lambda k * k.$$

This can be readily seen by taking the Fourier transform of both sides of the previous equation. The mathematical details proving convergence are covered in the paper by Delbracio et al. [1]. Finally, this last equation can be rewritten by using mostly zoom-in homothecies as

$$h = \lim_{n\to\infty} H_{\lambda^n} (k * H_{\frac{1}{\lambda}} k \ldots * H_{\frac{1}{\lambda^n}} k).$$

While mathematically equivalent in the continuous setting, performing the computations according to the last expression leads to a simpler implementation. Indeed, this permits to avoid the downsampling that would be required to deal with aliasing problems introduced when zooming-out digital images.

4 Algorithm Description

The **input** of the algorithm are the two digital images: $\tilde{\mathbf{v}}_1, \tilde{\mathbf{v}}_2$, the super-resolution factor s and the kernels (inter-image kernel and PSF) support size: $p \times q$ at the $s\times$ super-resolved grid.

The **output** of the algorithm are an $s\times$ sampling of the inter-image kernel $\mathbf{k}$ and an $s\times$ sampling of the camera PSF $\mathbf{h}$. Both images are of size $p \times q$.

The complete chain is summarized in the block diagram shown in figure 1.

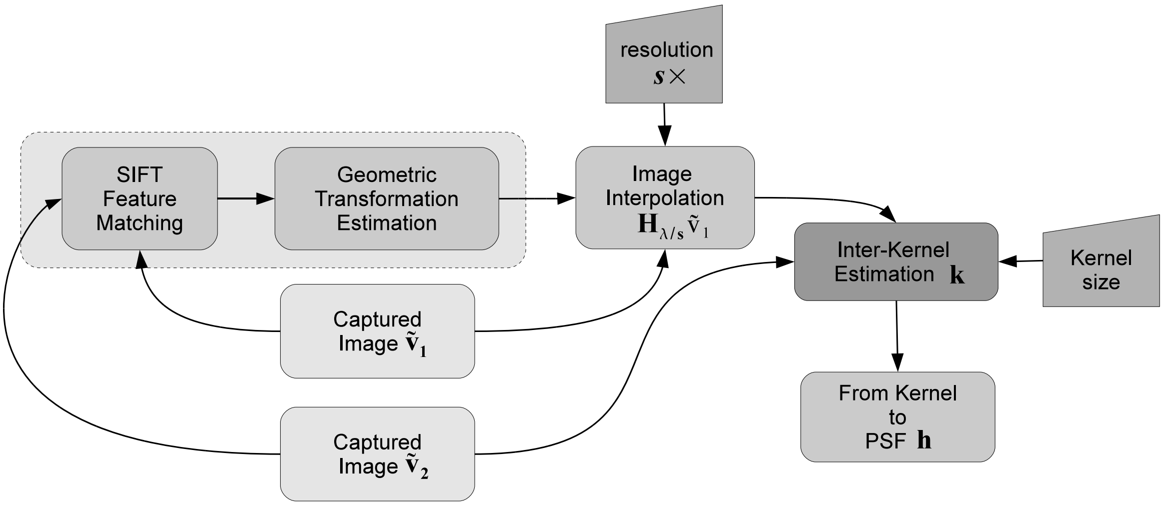

resolution
$s\times$
SIFT Feature Matching
Geometric Transformation Estimation
Image Interpolation $\mathbf{H}_{\lambda/s}\tilde{\mathbf{v}}_1$
Inter-Kernel Estimation $\mathbf{k}$
Kernel size
Captured Image $\tilde{\mathbf{v}}_1$
Captured Image $\tilde{\mathbf{v}}_2$
From Kernel to PSF $\mathbf{h}$

Figure 1: Block diagram summarizing the complete PSF estimation algorithm.

4.1 Image Subpixel Alignment, Geometric Transformation Estimation

In order to align both images and to estimate the geometric transformation from one to the other we use SIFT points and the ORSA-Homography subroutine described in the paper of Moisan et al. [4]. These subroutines may be replaced by any other accurate sub-pixel registration method. The important output of this stage is that given the two images we have a function D that maps one to the other. Suppose that the common parts of $\tilde{\mathbf{v}}_1$ and $\tilde{\mathbf{v}}_2$ are respectively of size $M_1 \times N_1$ and $M_2 \times N_2$. Then

$$D : [0, M_2 - 1] \times [0, N_2 - 1] \to [0, M_1 - 1] \times [0, N_1 - 1].$$

In the case of a homography, D can be expressed in homogeneous coordinates as a linear transform represented by the 3×3 matrix:

$$D = \begin{pmatrix} h_{0,0} & h_{0,1} & h_{0,2} \\ h_{1,0} & h_{1,1} & h_{1,2} \\ h_{2,0} & h_{2,1} & 1 \end{pmatrix}.$$

The common parts of the input images are found by computing the smallest rectangular sub-images containing all paired SIFT points. Additionally, the input images do not need to be given in the right order since from the estimated homography it can be easily deduced which one is the closest image and which one is the farthest one. From now on we will consider that $\tilde{\mathbf{v}}_1$ is the common region of the closest image and $\tilde{\mathbf{v}}_2$ the common region in the farthest one. Let $\mathbf{d}_1 = (d_{1_x}, d_{1_y})$ and $\mathbf{d}_2 = (d_{2_x}, d_{2_y})$ be the positions, in the original input full images, of the top left pixel of the extracted sub-images. These offsets are kept in order to directly work with the estimated geometric transform D, and the extracted sub-images.

4.2 Image Interpolation: $\mathbf{H}_{\lambda/s}\tilde{\mathbf{v}}_1$

In order to generate the rescaled samples $\mathbf{H}_{\lambda/s}\tilde{\mathbf{v}}_1$ we need to interpolate the digital image $\tilde{\mathbf{v}}_1$ at the desired scale λ/s. This is done by using the estimated geometric transformation D. From now on we will consider that 1× is the camera sampling frequency and the equivalent frequency band $[-\pi,\pi]^2$. The spectrum of the resampled image should be cut to be band-limited at $[-s\pi, s\pi]^2$ before resampling it. This is necessary to avoid aliasing artifacts. We do this by the following procedure:

1. Frequencies higher than $s\pi$ are cut-off from the closest image $\tilde{\mathbf{v}}_1$.

   (a) Compute the DCT (DCT-II from `libfftw` library) of $\tilde{\mathbf{v}}_1$. Let (c_{ij}), with $i = 0,\dots,M_1-1$ and $j = 0,\dots,N_1-1$ be the DCT coefficients of $\tilde{\mathbf{v}}_1$.

   (b) Set c_{ij} to zero for $i \geq sM_2$ or $j \geq sN_2$ (M_2 and N_2 are the number of rows and columns of the common region part in the captured zoom-out image $\tilde{\mathbf{v}}_2$).

   (c) Compute the inverse DCT (DCT-III from `libfftw` library) of (c_{ij}). Let $\bar{\mathbf{v}}_1$ be the resulting image.

2. Using the previously computed geometric transformation D, the filtered zoom-in image is interpolated at the desired resolution λ/s by bicubic interpolation (Keys parameter: -0.5). We proceed as follows:

   (a) Create a super-resolved $s\times$ regular sampling grid of $[0, M_2] \times [0, N_2]$, i.e., the step size is 1/s, first sample $(0,0)$. Let us denote this grid by g_s^2.

   (b) Transform the sampling grid by applying the estimated geometric transformation D to get $g_s^1 = D(g_s^2 + \mathbf{d}_2) - \mathbf{d}_1$.

   (c) Interpolate the values of $\bar{\mathbf{v}}_1$ at g_s^1 to get $\mathbf{H}_{\lambda/s}\tilde{\mathbf{v}}_1$. The size of this image is $(s(M_2-1)+1)\times(s(N_2-1)+1)$.

Additionally, in order to get rid of slight changes in illumination the mean pixel values of $\mathbf{H}_{\lambda/s}\tilde{\mathbf{v}}_1$ and $\tilde{\mathbf{v}}_2$ are subtracted from $\mathbf{H}_{\lambda/s}\tilde{\mathbf{v}}_1$ and $\tilde{\mathbf{v}}_2$ respectively.

In order to keep track of the common regions between the two images, an auxiliary binary image, that will serve as a mask, is computed. This mask is created by interpolating an image of the same size as $\tilde{\mathbf{v}}_2$, where each pixel value is 1, by the sampling grid g_s^1. Then, this mask is sub-sampled $s\times$ to get the 1× image mask $\mathbf{m}_2$. This mask will be necessary in the next section to restrict the estimation problem to the pixels that will be actually used.

4.3 Solving for the Inter-image Kernel

First, the following linear system is built:

$$\operatorname{argmin}_{\mathbf{k}} \quad \|\mathbf{M}\mathbf{S}_s\mathbf{U}\mathbf{k} - \mathbf{M}\tilde{\mathbf{v}}_2\|_2^2, \tag{3}$$

where the matrix $\mathbf{M}\mathbf{S}_s\mathbf{U}$ is composed of:

1. The matrix $\mathbf{U}$ associated to the 2D convolution operator with the interpolated image $\mathbf{H}_{\lambda/s}\tilde{\mathbf{v}}_1$. The convolution is done with a kernel of size $p \times q$.
2. The s-down-sampling matrix $\mathbf{S}_s$ takes the top left sample per each block of $s \times s$ pixels.
3. Finally a mask $\mathbf{M}$ is applied, setting to zero all values that are outside the *region of interest*.

The *region of interest* consists of the previously computed mask $\mathbf{m}_2$ that restricts the convolution to the intersection of both images. This mask is eroded by a square element of size $2r+1$, $r = \frac{\max(p,q)-1}{2s}$ to avoid boundary problems due to the convolution of finite support sequences. The farthest image $\tilde{\mathbf{v}}_2$ is reshaped as a vector to be consistent with the matrix formulation of the system, and then the matrix mask $\mathbf{M}$ is applied.

In practice the matrix $\mathbf{MS}_s\mathbf{U}$ is computed directly and its size is $\left(\frac{M+p-2}{s}+1\right) \times \left(\frac{N+p-2}{s}+1\right)$, $M \times N$ being the size of the image $\mathbf{H}_{\lambda/s}\tilde{\mathbf{v}}_1$. The reason for that is strictly the computational cost. Indeed, the sub-sampling operation consists of selecting only a few elements of the convolution with $\mathbf{H}_{\lambda/s}\tilde{\mathbf{v}}_1$. Thus, we restrict the computation to those elements that will result from the sub-sampling operation. The product of the resulting subsampling $\mathbf{S}_s\mathbf{U}$ with the mask $\mathbf{M}$ follows the same idea: if an element of $\mathbf{S}_s\mathbf{U}$ is selected by the mask, its value is kept unchanged; otherwise, it is set to zero, meaning that will not be taken into account further.

Problem (3) is finally solved by a least squares algorithm.

4.4 From the Inter-image Kernel to the PSF

In order to recover the camera PSF h we need to compute (see the article of Delbracio et al. [1])

$$h = \lim_{n\to\infty} H_{\lambda^n}(k * H_{\frac{1}{\lambda}}k \ldots * H_{\frac{1}{\lambda^n}}k).$$

The value $\lambda = (\lambda_x, \lambda_y)$ is estimated from the geometric transformation D. In the case D is estimated as a homography, the scale values are taken as $\lambda = (h_{0,0}, h_{1,1})$. This corresponds to the situation where D is a pure zoom, and is a good approximation to the fronto-parallel acquisition with negligible rotation. We proceed as follows:

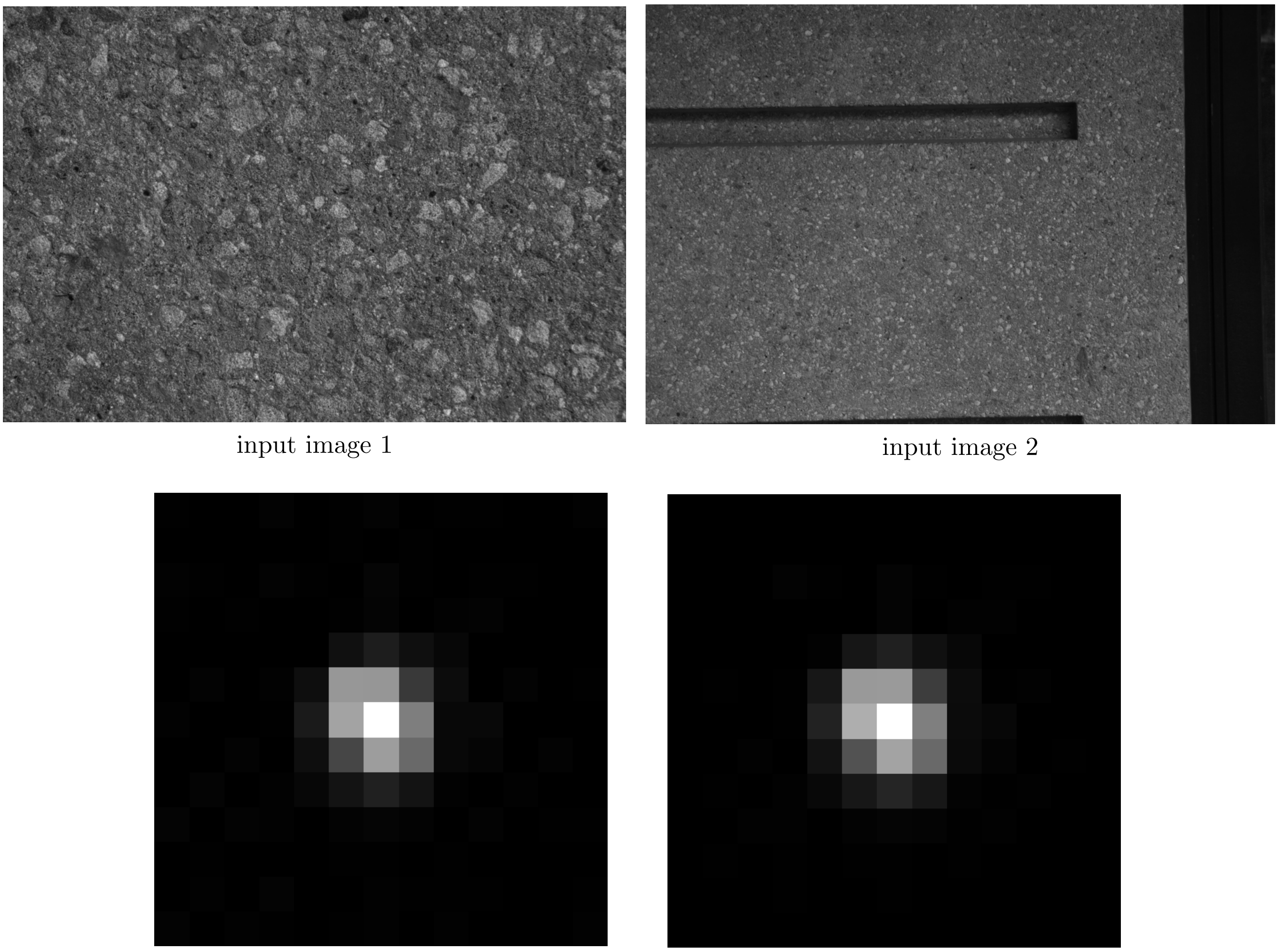
1. Initialize $\mathbf{u}_0 = \mathbf{k}$, $n = 1$.
2. Compute $\mathbf{H}_{1/\lambda^n}\mathbf{k}$ by using $\lambda = (\lambda_x, \lambda_y)$.
   (a) Let $p_n = \lceil p\lambda_y^n \rceil$ and $q_n = \lceil q\lambda_x^n \rceil$ be the nearest larger integers to $p\lambda_y^n$ and $q\lambda_x^n$ respectively. We create a sampling grid with step $(1/\lambda_x^n, 1/\lambda_y^n)$ of size $p_n \times q_n$. This grid should be centered at $x_c = (p-1)/2$, $y_c = (q-1)/2$. This is done by considering sample positions $(x_{ij}, y_{ij}) = (j/\lambda_x + \tau_x\,,\, i/\lambda_y + \tau_y)$, with $i = 0, \ldots, p_n - 1$ and $j = 0, \ldots, q_n - 1$. The offset $(\tau_x, \tau_y) = \left(\frac{q-1}{2} - \frac{q_n-1}{2\lambda_x}, \frac{p-1}{2} - \frac{p_n-1}{2\lambda_y}\right)$ is necessary to center the grid.
   (b) Interpolate, by bicubic interpolation (Keys parameter -0.5), the values of $\mathbf{k}$ to get $\mathbf{H}_{1/\lambda^n}\mathbf{k}$ using the sampling grid defined in the previous step.
3. Calculate $\mathbf{u}_n = \mathbf{H}_{1/\lambda^n}\mathbf{k} * \mathbf{u}_{n-1}$. This image has the same size as $\mathbf{H}_{1/\lambda^n}\mathbf{k}$ and is the central part of the convolution coinciding with the support of $\mathbf{H}_{1/\lambda^n}\mathbf{k}$.
4. If $\min\{\lambda_x^n, \lambda_y^n\} > \lambda_{\max}$ or $n = n_{\max}$ go to 5. Else update $n := n+1$ and repeat from 2.
5. Calculate $\mathbf{h} = \mathbf{H}_{\lambda^n}\mathbf{u}_n$.
   (a) We create a sampling grid with step $(\lambda_x^n, \lambda_y^n)$ of size $p \times q$. This grid should be centered at $x_c = (p-1)/2$, $y_c = (q-1)/2$. This is done by considering sample positions $(x_{ij}, y_{ij}) = (j\lambda_x^n + \tau_x\,,\, i\lambda_y^n + \tau_y)$, with $i = 0, \ldots, p-1$ and $j = 0, \ldots, q-1$. The offset $(\tau_x, \tau_y) = \left(\frac{q_n-1}{2} - \frac{q-1}{2}\lambda_x, \frac{p_n-1}{2} - \frac{p-1}{2}\lambda_y\right)$ is necessary to center the grid.
   (b) Interpolate, by bicubic interpolation (Keys parameter -0.5), the values of $\mathbf{H}_{\lambda^n}\mathbf{u}$ to get $\mathbf{h}$ using the sampling grid defined in the previous step.

The algorithm converges after a few iterations since λ^n grows very fast. We set $n_{\max} = 3$ and $\lambda_{\max} = 50$, since the convolution with the inter-image kernel zoomed-out 50× or greater produces a negligible change in the final result.

Thresholding negative values. Since negative light does not exist the estimated PSF should be positive. We can therefore constrain the solution to be non-negative by projecting the result of step 5 to the non-negative half-space.

5 A Running Example

In this section we show an example of the algorithm applied to a PSF estimation at 3× the sensor resolution from real camera acquisitions. Figure 2 shows two digital images acquired by a Canon EOS 400D camera provided with a Tamron AF 17-50mm F/2.8 XR Di-II lens, and the PSF estimation from the inter image kernel. The estimation was done for the blue channel. Figure 3 shows the registered and interpolated closest image, necessary for the estimation of the inter-image kernel. Also, the mask showing the pixels that are actually used for the estimation is presented. See caption of figure 3 for details.



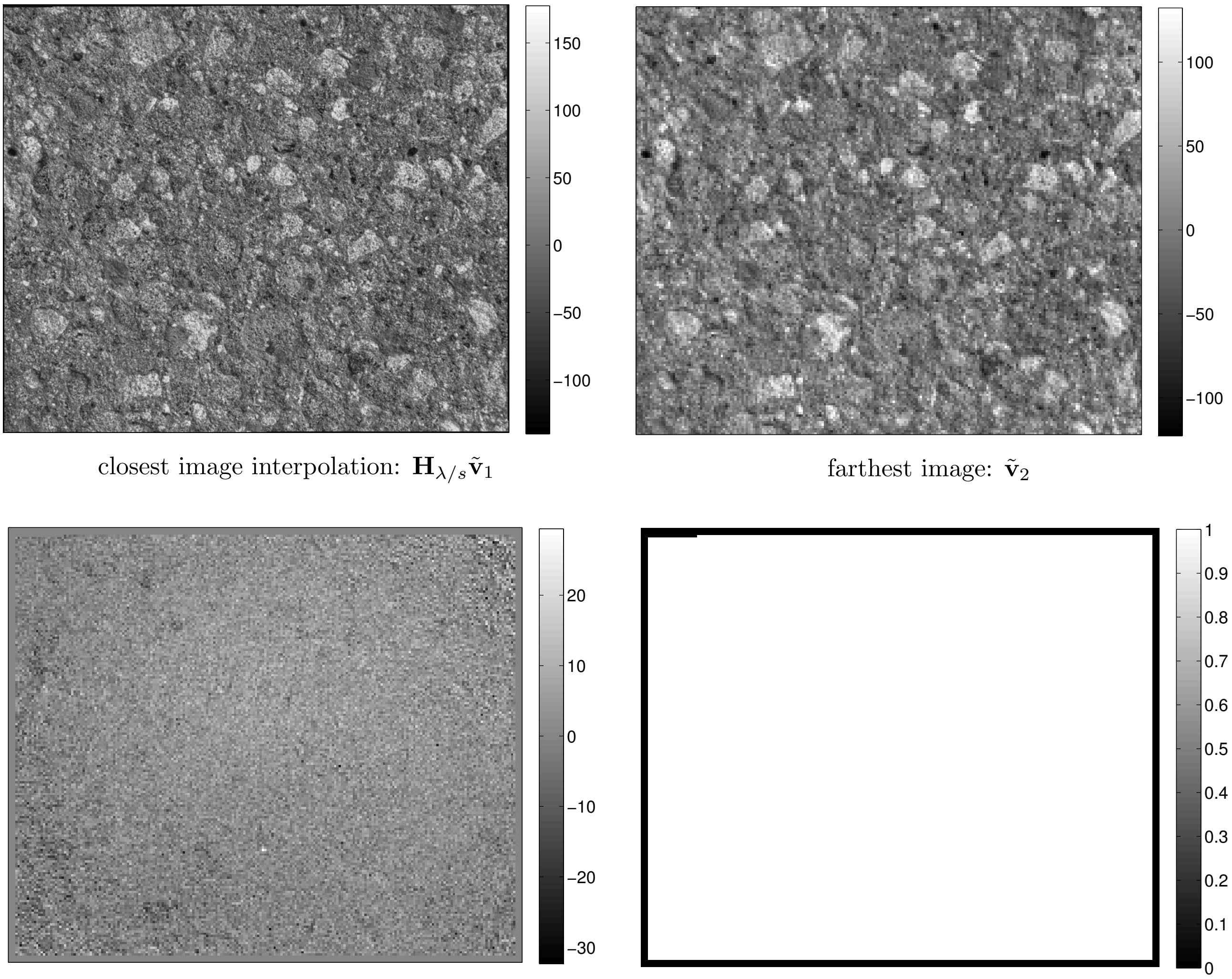
input image 1

input image 2

3× inter image kernel estimation

3× PSF estimation

Figure 2: An example of a pair of digital images that allow to estimate the PSF. Top: two distant, parallel views of a wall. Bottom: the *inter-image kernel* between these two views. The inter-image kernel models the necessary blur that should be applied to the closest image to produce the farthest image (with the necessary zooming). The estimated inter-image kernel and camera PSF are obtained at 3× the camera resolution for the blue channel. Although no regularization is imposed, the kernels are smooth.



closest image interpolation: $\mathbf{H}_{\lambda/s}\tilde{\mathbf{v}}_1$

farthest image: $\tilde{\mathbf{v}}_2$

residual image

estimation mask

Figure 3: Top row: closest sub-image re-interpolated to generate $\mathbf{H}_{\lambda/s}\tilde{\mathbf{v}}_1$and the original farthest sub-image $\tilde{\mathbf{v}}_2$. Bottom row: the residual image $\mathcal{S}_s(\mathbf{H}_{\frac{\lambda}{s}}\tilde{\mathbf{v}}_1 * \mathbf{k}) - \tilde{\mathbf{v}}_2$, and the mask used for the estimation.

6 Conclusion

In this work we presented an algorithm for the subpixel estimation of the point spread function of a digital camera from aliased photographs. The procedure is based on taking two fronto-parallel photographs of the same flat textured scene, from different distances leading to different geometric scales, and then estimating the kernel blur between them. The proposed algorithm does not require any special acquisition setup or calibration pattern.

7 Appendix: General Tips for the Set-up

1. The scene should be as planar and as textured as possible.
2. The photographs should be taken between 2 and 8 relative distance. The possible super-resolution factor is always less than the relative distance, so for 3× estimation relative distance should be higher than 3.
3. To produce accurate estimations it is highly recommended to use a tripod to avoid handheld

shake.

4. Both photographs should be taken with the same camera parameters. The only exception is camera focus, that should be re-set to have both images in focus.
5. The images should be taken with the same illumination conditions.
6. Both images should be recorded in RAW format (no compression, no post-processing: no demosaicking, no denoising, no enhancing, etc). RAW conversion is camera-dependent and not provided by our demo. In our examples a suitable conversion of the RAW format to a PGM image containing the Bayer pattern could be achieved by the command "dcraw -4 -d input.raw output.pgm". A single color channel should then be extracted and used as the input to our algorithm.

Acknowledgements

Research partially funded by the Centre National d'Etudes Spatiales (R&T), the European Research Council advanced grant "Twelve Labours", the Office of Naval Research (grant N00014-97-1-0839), STIC-AmSud (11STIC-01 - MMVPSCV), and the Uruguayan Agency for Research and Innovation (ANII) under grant PR-POS-2008-003.

Image Credits

References

[1] M. Delbracio, A. Almansa, J. Morel, and P. Musé. Subpixel point spread function estimation from two photographs at different distances. *SIAM Journal on Imaging Sciences*, 5(4):1234–1260, 2012. http://dx.doi.org/10.1137/110848335.

[2] M. Delbracio, P. Musé, A. Almansa, and JM. Morel. The non-parametric sub-pixel local point spread function estimation is a well posed problem. *International Journal of Computer Vision*, 96:175–194, 2012. http://dx.doi.org/10.1007/s11263-011-0460-0.

[3] N. Joshi, R. Szeliski, and D. Kriegman. PSF estimation using sharp edge prediction. In *IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, Alaska, June 2008. http://dx.doi.org/10.1109/CVPR.2008.4587834.

[4] Lionel Moisan, Pierre Moulon, Pascal Monasse. Automatic Homographic Registration of a Pair of Images, with A Contrario Elimination of Outliers. *Image Processing On Line*, 2012. http://dx.doi.org/10.5201/ipol.2012.mmm-oh.