



Published in Image Processing On Line on 2012-03-23.  
 Submitted on 2012-00-00, accepted on 2012-00-00.  
 ISSN 2105-1232 © 2012 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.2012.admm-nppsf>

# Non-parametric Sub-pixel Local Point Spread Function Estimation

Mauricio Delbracio<sup>1</sup>, Pablo Musé<sup>2</sup>, Andrés Almansa<sup>3</sup>

<sup>1</sup>CMLA, ENS Cachan, France (mdelbra@fing.edu.uy)

<sup>2</sup>IIE, UdelaR, Uruguay (pmusel@fing.edu.uy)

<sup>3</sup>TELECOM ParisTech, France (andres.almansa@telecom-paristech.fr)

*Communicated by* Sylvain Durand

*Demo edited by* Mauricio Delbracio

## Abstract

This work presents an algorithm for the local subpixel estimation of the Point Spread Function (PSF) that models the intrinsic camera blur. For this purpose, the Bernoulli(0.5) random noise calibration pattern introduced in a previous article [1] is used. This leads to a well-posed near-optimal accurate estimation. First the pattern position and its illumination conditions are accurately estimated. This allows for accurate geometric registration and radiometric correction. Once these procedures are performed, the local PSF can be directly computed by inverting a linear system. This system is well-posed and consequently its inversion does not require any regularization or prior model. The PSF estimates reach stringent accuracy levels with a relative error in the order of 2 to 5%.

## Source Code

ANSI C source code to produce the same results as the demo and a vector image of the proposed calibration pattern are accessible on the article web page <https://doi.org/10.5201/ipol.2012.admm-nppsf>. Future software releases and updates will be posted at <https://github.com/mdelbra/psf-estim>.

**Keywords:** image blur, subpixel convolution kernel estimation; aliasing, inverse problems, camera quality assessment, point spread function, modulated transfer function

## 1 Overview

Extrinsic image blur can be observed when the camera's focus is not correctly adjusted by the user, when the objects in the scene appear at different depths, or when the relative motion between the camera and the scene is faster than the shutter speed (motion blur). Besides these sources of blur, even in ideal acquisition conditions there is a permanent intrinsic physical camera blur due to light diffraction, sensor resolution, lens aberration, and anti-aliasing filters. Our goal here is to accurately

estimate the Point Spread Function (PSF), that models the intrinsic camera blur. This function can be locally interpreted as the response of the camera to a point light source.

Most medium to high quality digital cameras acquire images at a spatial rate which is below the Nyquist rate. For this reason only aliased versions of the camera point-spread function (PSF) can be directly observed. Yet, it can be recovered, at a sub-pixel resolution, by a numerical method. Since the acquisition system is only locally stationary, this PSF estimation must be local.

In a previous article [1] we presented a theoretical study proving that the sub-pixel PSF estimation problem is well-posed even with a single image capture, as long as the captured scene is well chosen. Indeed, theoretical bounds show that a near-optimal accuracy can be achieved by taking a single snapshot of a calibration pattern mimicking a Bernoulli(0.5) white noise. We first use an algorithm to accurately estimate the pattern position and its illumination conditions. This allows for accurate geometric registration and radiometric correction; Once these procedures have been applied, the local PSF can be directly computed by inverting a linear system that is well-posed and consequently its inversion does not require any regularization or prior model. The PSF estimates reach stringent accuracy levels with a relative error in the order of 2 to 5%. To the best of our knowledge, such a regularization-free and model-free sub-pixel PSF estimation scheme is the first of its kind.

This article is organized as follows. Section 2 describes the general mathematical digital camera model used for PSF estimation method. Section 3 introduces the proposed PSF calibration pattern and some tips concerning the set-up. In Section 4, a detailed description of the implemented algorithm is given. In Section 5, experimental results generated with real camera data are presented. An ANSI C implementation and a vector image of the proposed calibration pattern can be downloaded from the article web page <https://doi.org/10.5201/ipol.2012.admm-nppsf>. Future software releases and updates will be posted at <https://github.com/mdelbra/psf-estim>.

## 2 The Camera Model

An accurate estimation of the PSF requires a proper modeling of the digital image formation process. The basic pin-hole camera model consists of a perspective projection of the 3D world scene into the focal plane. In real cameras, a system of lenses is needed to concentrate the light rays toward the focal point, passing through a finite but non pin-hole aperture. Hence, the perspective projection is followed by geometric distortions, which are always present in any camera/lens system. This process can be faithfully modeled as a diffeomorphism from the focal plane into itself. The blur of the resulting image in the focal plane is modeled by a kernel that captures all PSF like effects (diffraction due to finite aperture, lens aberration, optical low-pass filter, sensor light integration, etc). Finally the resulting analog image is sampled into a discrete image by the sensor array.

If we consider that the observed scene is a planar scene  $u$ , the perspective projection is reduced to a planar homography that will be denoted by  $H$ . The whole image formation process can therefore be summarized in a single equation:

$$\mathbf{v} = \mathbf{S}_1 (g((u \circ H \circ F) * h)) + \mathbf{n}, \quad (\text{M}')$$

where  $F(\cdot)$  is the geometric distortion field,  $h$  is the convolution kernel due to all PSF like effects and  $g(\cdot)$  is a monotone non-decreasing function that describes the non-linear sensor response (camera response function - CRF). The operator  $\mathbf{S}_1$  is the bi-dimensional ideal sampling operator due to the sensor array, and  $\mathbf{n}$  models the sensor noise.

The blur kernel  $h$  is space variant, but evolves smoothly. Thus, the symbol  $*$  is understood as a local convolution product, the kernel  $h$  varying smoothly with the position in the image domain.

The model can be further simplified by noticing that, in order to estimate  $h$ , the geometric transformation  $D = H \circ F$  can be considered as a whole: there is no need to estimate separately the

projective and non-projective parts. We shall therefore denote by  $D$  the whole geometric transformation, and the image formation model finally is

$$\mathbf{v} = \mathbf{S}_1 g(u_D * h) + \mathbf{n}, \quad (\text{M}''')$$

where  $u_D$  is the geometrically transformed image, namely  $u_D(\mathbf{x}) = u(D(\mathbf{x}))$ . This model can be further simplified. Indeed the sampling and the contrast change  $g$  commute, so that

$$\mathbf{S}_1 g(u_D * h) = g(\mathbf{S}_1 u_D * h).$$

As we shall see, the contrast change  $g$  can be recovered from the image samples. Thus we will first focus on the simplified formation model

$$\mathbf{v} = \mathbf{S}_1(u_D * h) + \mathbf{n}, \quad (\text{M})$$

and explain later on how the  $g$  term can be eliminated.

## 2.1 The Digital Camera Model

In the following,  $\hat{f}$  denotes the Fourier Transform of a function  $f$ . The s-Shannon-Whittaker interpolator  $I_s$  is defined as

$$I_s \mathbf{u}(\mathbf{x}) = \sum_{\mathbf{k} \in \mathbb{Z}} \mathbf{u}(\mathbf{k}) \text{sinc}(\mathbf{x}/s - \mathbf{k})$$

while  $\mathbf{S}_s$  is the s-over-sampling operator

$$\mathbf{S}_s u(\mathbf{k}) = u(s^{-1} \mathbf{k})$$

for  $\mathbf{k} \in \mathbb{Z}$  and  $\text{LPF}_w$  is the frequency cut-off low pass filter that cuts the spectrum of a signal to  $[-w\pi, w\pi]$ . Let us also denote by  $\mathcal{S}_s$  the s-to-1-sub-sampling operator  $\mathcal{S}_s = \mathbf{S}_1 I_s$ .

Suppose that  $h$  is band-limited within  $\text{supp}(\hat{h}) = [-\delta\pi, \delta\pi]$ . If the PSF is well-sampled (i.e. sampled at a rate  $s$ , where  $s > \delta$ ), the Nyquist sampling theorem guarantees a perfect signal reconstruction. We will concentrate on the case where  $\delta > 1$ , which corresponds to aliased images, as in practice most digital cameras introduce aliasing. Moreover, if the PSF is well-sampled the continuous image formation model can be written in terms of discrete sequences. The sampling of the continuous convolution between the scene  $u_D(x)$  and the kernel  $h$  can be written as the discrete convolution of the samples of  $\tilde{u}_D = \text{LPF}_s u_D$  and the samples of  $h$  as shown in the original paper [1] (Lemma 1). Next,

$$\mathbf{v} = \mathcal{S}_s \tilde{\mathbf{u}}_D * \mathbf{h} + \mathbf{n},$$

where  $\mathbf{h}$  and  $\tilde{\mathbf{u}}_D$  are sampled at rate  $s$  such that  $s > \delta$  for  $\mathbf{h}$  to be well sampled. Recall that  $\tilde{\mathbf{u}}_D$  is the result of the Shannon-sampling on the  $s \times$  grid of the distorted continuous filtered pattern  $\tilde{\mathbf{u}}_D = \mathbf{S}_s \text{LPF}_s u_D$ , and  $\mathbf{v}$  the blurred degraded digital observation on the camera  $1 \times$  sensor grid.

The value  $s$  is the super-resolution sampling rate in the high resolution lattice, where the PSF estimation is going to take place, from the  $1 \times$  sensor grid.

## 2.2 Inverse Problem Statement in Terms of the Re-sampling Operator and the Calibration Pattern

Assume we can unveil exactly the latent sharp image that produced the blurry aliased observation. Then, solving for the PSF amounts to solve an inverse problem governed by the image formation model (M). The first step toward solving this problem is to carefully model the re-sampling operator that produced an aliased observation. The inverse problem to be solved can be stated in terms of the re-sampling rate and of the observed pattern image  $u_D$ , which is a function of the calibration pattern. It follows, as will soon become clear, that the accuracy of the estimation of  $h$  depends on how well we can invert an operator that depends on the re-sampling operator and on the calibration pattern.

As shown in the original work [1], nearly optimal conditioning can be obtained when the calibration pattern is a realization of a white noise process. While this may not be new (noise patterns have been previously used in non-blind PSF estimation), the novelty presented is that the use of white noise patterns allows one to solve for super-resolved PSFs without the need for any regularization, and without any prior model for  $h$ . In other words, the system is well posed as long a white noise image is chosen as the calibration pattern.

Assuming that  $\mathbf{n}$  is a zero-mean stationary white Gaussian noise, the kernel samples  $\mathbf{h}$  can be obtained by solving

$$\operatorname{argmin}_{\mathbf{h}} \quad \|\mathcal{S}_s \tilde{\mathbf{u}}_D * \mathbf{h} - \mathbf{v}\|_2^2. \quad (\text{P}')$$

As inferred by the above discussion, to estimate the PSF by a non-blind method one has to address the following issues:

- choice of a good PSF calibration pattern;
- estimation of the function  $g(\cdot)$ , the non-linear sensor response;
- estimation of the geometric deformation  $D(\cdot)$ ;
- construction of  $\tilde{\mathbf{u}}_D$  from the sharp latent pattern image  $u$ ;
- numerical estimation of the PSF.

So far  $h$  is only assumed to be band-limited. The numerical method will recover only a finite number of samples of  $h$ , which is well localized, and therefore in practice compactly supported in space. Strictly speaking  $h$  being band-limited cannot be compactly supported. However, the error introduced by a constraint on the support will prove negligible in comparison to the other sources of error: image noise, quantization, slight estimation errors of  $g$ ,  $D$ , ... From now on, we choose the support of  $\mathbf{h}$  to be a square patch of size  $r \times r$ . In practice, this will be the only free parameter of the method. However, experimentally the obtained solution for the superresolved kernel  $\mathbf{h}$  proved to be independent from variations of its assumed support as long as  $r$  is within a reasonable range (typically between  $3s + 1$  and  $4s + 1$ ).

The problem (P') can be rewritten in matrix form,

$$\operatorname{argmin}_{\mathbf{h}} \quad \|\mathbf{S}_s \mathbf{U} \mathbf{h} - \mathbf{v}\|_2^2, \quad (\text{P})$$

where  $\mathbf{U}$  is the convolution matrix by  $\tilde{\mathbf{u}}_D$ . (This matrix is applied to the sample vector  $\mathbf{h}$ .) Assuming that the observed image  $\mathbf{v}$  is of size  $m \times n$ , the size of  $\tilde{\mathbf{u}}_D$  becomes  $ms \times ns$ . The matrix  $\mathbf{S}_s$  is the downsampling matrix of size  $M \times Ms^2$ , where  $M = m \times n$ . As mentioned above, we need  $s > \delta$

to recover  $h$  from its samples. Thus,  $s$  is an integer greater than  $\delta$ , which facilitates the construction of the subsampling matrix  $(\mathbf{S}_s \mathbf{u})(m, n) = \mathbf{u}(ms, ns)$ . Then  $\mathbf{S}_s \mathbf{U}$  is of size  $M \times N$ , with  $N = r \times r$ .

The solution of ((P)) is easily obtained using a least squares estimation procedure, and is given by

$$\mathbf{h}_e = (\mathbf{S}_s \mathbf{U})^+ \mathbf{v},$$

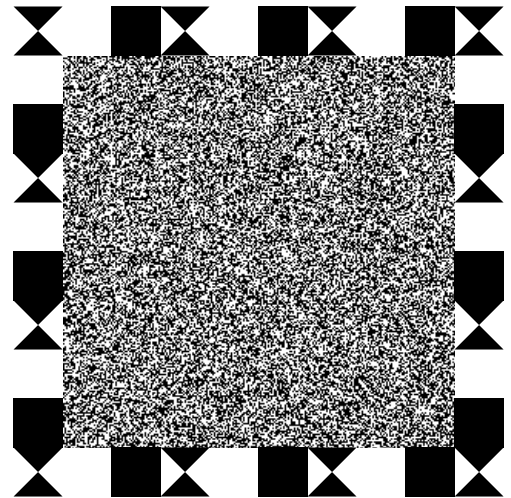
where  $(\mathbf{S}_s \mathbf{U})^+ = ((\mathbf{S}_s \mathbf{U})^t (\mathbf{S}_s \mathbf{U}))^{-1} (\mathbf{S}_s \mathbf{U})^t$  is the Moore-Penrose pseudo-inverse of  $(\mathbf{S}_s \mathbf{U})$ . If matrix  $(\mathbf{S}_s \mathbf{U})$  is full rank, the problem (P) admits a unique solution. However, numerically the inversion will be well-posed or not depending on the condition number of this matrix.

Light diffraction, optical low pas filtering, and sensor light integration all produce non-negative kernels. Consequently the estimated PSF must be non-negative. We can therefore introduce a non-negativity constraint on the solution, thus reducing the feasibility space. Not imposing this non-negativity assumption yields essentially the same results (see the examples section), which in fact verifies the correctness of the proposed image formation model. Nevertheless, if one still wants to force the solution to be non-negative, it can be obtained by a non-negative least squares procedure, or by simply thresholding the unconstrained least squares solution to discard negative values.

### 3 The Pattern Set-up

#### 3.1 Structure of the Pattern

The pattern consists of  $256 \times 256$  small black/white squares drawn independently from a Bernoulli(0.5) distribution. This image is surrounded by checkerboard-like marks and black/white squares for alignment and illumination estimation purposes. The surrounding black and white squares in the checkerboard like frame are 32 times larger than a point (actually a square) in the noise region. A vector image file of this calibration pattern is included as Supplementary Material and can be downloaded from the article web page <https://doi.org/10.5201/ipol.2012.admm-nppsf>.



*Calibration pattern*

#### 3.2 Tips for the Set-up

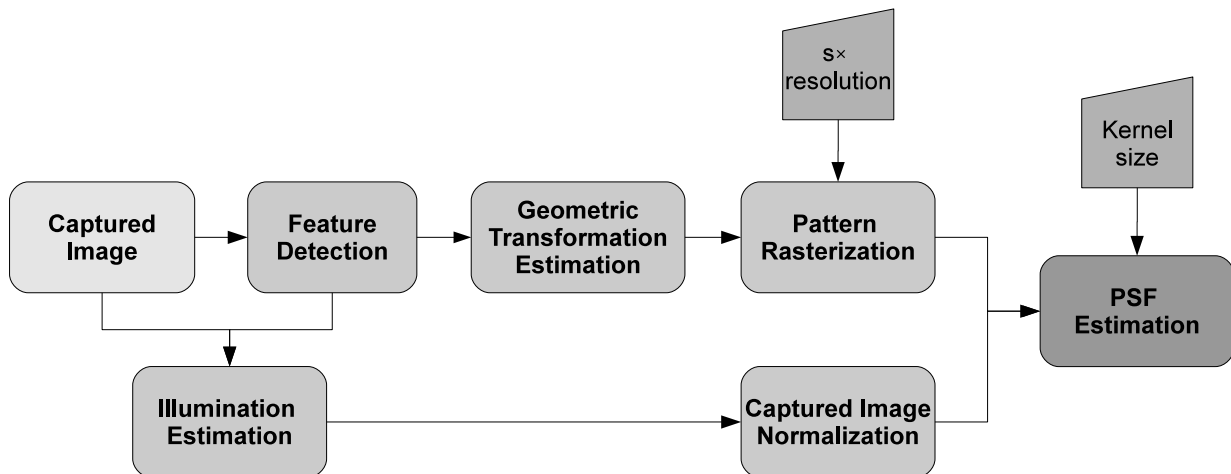
- The pattern should be printed at a size/resolution/quality such that the small black and white squares are effectively squares and the transition edge between them is completely negligible. For all the experiments shown here the pattern size was 30cm x 30cm. Beware that by default most printers process the image during the printing to avoid aliasing artifacts. Be careful to turn off this pre-processing as much as possible.
- The printed pattern should be carefully pasted or printed on top of a surface as planar as possible (a mirror, for instance).

- The pattern should be photographed to cover only  $100 \times 100$  pixels approximately (it has been designed to maximize its effectiveness at these values).
- Although the algorithm takes care of small variations in light, illumination with reasonably good uniformity is indispensable for producing accurate estimations.

Other ideas on how to build a camera calibration lab can be found on: “The Imatest Test Lab - How to build a testing lab<sup>1</sup>”.

## 4 Algorithm

The complete chain is summarized in the block diagram of the following figure.



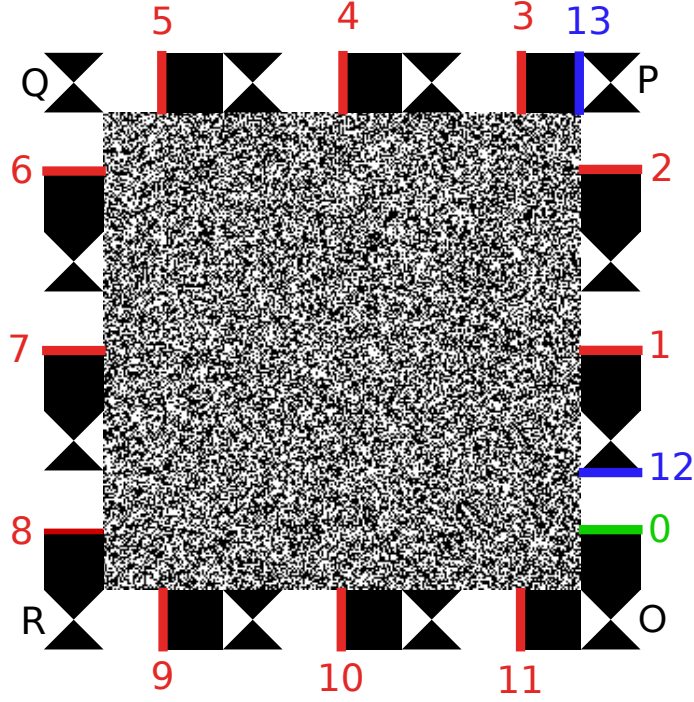
The captured image is precisely aligned to the analytic pattern by means of the surrounding checkerboard markers. Non-uniform illumination and non-linear camera response function impact - CRF are corrected from the captured image to allow an artifact-free superresolved PSF estimation.

The next paragraphs present a brief summary for each block.

### 4.1 Pattern Detection

In order to detect the pattern in the image we use the line segment detector LSD algorithm by Grompone et al. [2]. The idea is to detect the segments that are present in the pattern structure as can be seen in the following figure.

<sup>1</sup><http://www.imatest.com/docs/lab.html>



By using the detected line segments (here shown in red, green and blue for illustration purpose) we can approximate the checkerboard corners. These initial corners locations are re-adjusted by a subsequent subpixel stage. The pattern detection procedure can be summarized into the following steps:

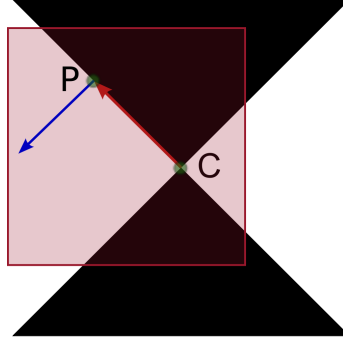
1. Detect all lines segments in the image. Each segment is represented by its two extreme points.
2. For each detected segment  $s_i$  do:
  - (a) Take another detected segment  $s_j$ ,  $j \neq i$ .
  - (b) Calculate  $s_{ij}$ , the coordinates of segment  $s_j$  in a new coordinate system relative to segment  $s_i$ . This new coordinate system maps the first extreme point of segment  $s_i$  to  $(0,0)$  and the second one to  $(0,1)$ . The idea is to detect all the red and blue segments in relative position to the green one.
  - (c) Check if  $s_{ij}$  is one of the searched segments:
    - i. Compute the distance between  $s_{ij}$  and the ideal relative position of every searched segments (red and blue segments).
    - ii. If it is up to a tolerance and the distance is less than in the previous detection of the same segment, update the segment and the distance to the optimal position
3. If there is one and only one segment that has all associated segments we consider that the pattern has been detected.

Once the pattern with the aforementioned segments has been detected, an initial estimate of the positions of O, P, Q, R is computed as the middle points of segments 1,4,7 and 10.

## 4.2 Pattern Subpixel Alignment

In order to deal with geometric distortions the ideal pattern and its observation have to be precisely aligned locally. For that reason we have introduced checkerboard corners. Several methods to detect checkerboard corners have been reported in the Computer Vision literature ranging from differential operators such as Harris detector to more specific correlation methods. In this work we used a Harris-Stephens based corner detector in which we iteratively refine the detected corner positions to reach subpixel accuracy. The present algorithm is included in the [OpenCV library](#)<sup>2</sup>.

The subpixel corner detector is based on the fact that the image gradient at a point close to the center is orthogonal to every vector from the center to that point.



This can be mathematically expressed as

$$\nabla I_P^t \cdot CP = 0$$

for every  $P$  in a neighborhood of  $C$ . As in practice the acquired image will be contaminated with noise, this expression will not be strictly zero. This last expression can be written as  $(\nabla I_P \cdot \nabla I_P^t) \cdot (P - C) = 0$  and since this expression holds for  $P$  in a neighborhood of  $C$ ,

$$\sum_{P \in \mathcal{N}(C)} (\nabla I_P \cdot \nabla I_P^t) \cdot P = \sum_{P \in \mathcal{N}(C)} (\nabla I_P \cdot \nabla I_P^t) \cdot C$$

In order to give more importance to points closer to the center  $C$ , a weighting function  $W_P = f(\|P - C\|)$  is included. Then,  $C$  is found by least squares

$$C = \left( \sum_{P \in \mathcal{N}(C)} (W_P \nabla I_P \cdot \nabla I_P^t) \right)^{-1} \left( \sum_{P \in \mathcal{N}(C)} (W_P \nabla I_P \cdot \nabla I_P^t) \cdot P \right). \quad (1)$$

Based on this idea, the checkerboard detector algorithm iteratively runs as follows:

1. Given  $C$  (with possible subpixel precision), compute  $\mathcal{N}(C)$  as a square of size  $2R + 1 \times 2R + 1$  centered in  $C$ . The non-integer pixel values are calculated using bilinear interpolation.
2. The gradient  $\nabla I_P$  is calculated by finite differences in every point  $P$  of  $\mathcal{N}(C)$ .
3.  $\nabla I_P \cdot \nabla I_P^t W_P$  is computed where  $W_P = f(\|P - C\|)$  is an isotropic Gaussian function centered in  $C$ . The variance of the Gaussian function is  $\sigma_{W_P}^2 = \frac{(2R+1)^2}{2}$ .
4. Set  $C_{\text{old}} = C$  and compute the new  $C$  by solving Equation (1).
5. If  $\|C - C_{\text{old}}\| < \text{tol}$  or  $++\text{iter} > \text{max\_iter}$  exit, else go to 1.

For all our experiments we set  $\text{tol} = 10^{-5}$ ,  $\text{max\_iter} = 200$ ,  $R = 3$ .

<sup>2</sup><http://opencv.willowgarage.com/wiki/>



### 4.3 Geometric Transformation Estimation

For our purpose of PSF estimation we do not need to decompose the distortion into its homography and non-homography parts, as it is done in classical geometric camera calibration. Instead we use thin-plate splines [3] to model the whole deformation. Since we have previously detected the  $\{\tilde{p}_i\}$  checkerboard corners centers from the observed target image and we know exactly their ideal corresponding locations  $\{P_i\}$ , we can use these correspondences to find a smooth mapping from the non-distorted to the distorted space.

The thin-plates are found by minimizing the functional:

$$E = \sum_i \|f(P_i) - \tilde{p}_i\|^2 + \lambda \iint (f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2) dx dy.$$

where  $\lambda$  controls the amount of regularization. As we will show if  $\lambda$  is zero the thin-plates interpolate the example points and if it goes to infinity the mapping becomes a pure affine transformation.

The solution of this functional is of the form

$$f_i(x, y) = d_{i0} + d_{i1}x + d_{i2}y + \sum_{j=1}^n c_{ij}\Phi(\|P_j - (x, y)\|) \quad i = 1, 2,$$

where  $\Phi(\mathbf{x}) = \|\mathbf{x}\|^2 \log \|\mathbf{x}\|$ ,  $n$  is the number of points (in our case  $n$  is the number of checkerboard corners i.e. 12). A nice property of the thin-plates is that it can always be decomposed into an affine and a non-affine component. The 6 coefficients  $(d_{ij})$  form the affine part and the  $n \times 2$  coefficients  $(c_{ij})$  form the non-affine part. We now show how both set of coefficients can be obtained. In the following we represent points in homogeneous coordinates  $(x, y, 1)$ . Let us call  $Y$  and  $X$  the  $n \times 3$  concatenated versions of the point coordinates  $P_i$  and  $\tilde{p}_i$  respectively. We denote by  $\Phi$  the  $n \times n$  matrix formed from  $\Phi(\|P_i - \tilde{p}_j\|)$ ,  $i, j = 1..n$ , representing the thin-plate kernel.

It can be shown (see Sprengel et al. [3]) that the solution is given by

$$\begin{aligned} \hat{c} &= Q_2(Q_2^T \Phi Q_2 + \lambda I_{(n-3)})^{-1} Q_2^T Y \\ \hat{d} &= R^{-1} Q_1^T (Y - \Phi \hat{c}), \end{aligned}$$

where  $Q_1$  and  $Q_2$  are obtained from the QR decomposition of matrix  $X$ ,

$$X = [ Q_1 \quad Q_2 ] \begin{bmatrix} R \\ 0 \end{bmatrix}.$$

In this application as we always have a fixed number of points – 12 – and the support of the pattern in the image is very small (about  $100 \times 100$  pixels), the geometrical distortion is expected to be minimal and thus very well approximated by an affinity. In practice, setting  $\lambda = 10$  proved to be an appropriate trade-off.

### 4.4 Illumination Estimation and Normalization

In order to solve the minimization problem (P), the gray levels in the sharp image pattern and those in the observed image have to be matched. For that purpose, the proposed pattern has black and white square regions to estimate the mapping between black and white colors and the respective observed gray values. As there are several of these constant regions located at different places, we can estimate a black (white) image that models the black (white) intensity level at each pixel. We model this black (white) image by a second order polynomial:

$$I_{\text{black}}(x, y) = ax^2 + by^2 + cx + dx + e.$$

The estimation of the coefficients is done by least squares from the known pairs (value, position). Then, assume  $v(x, y)$  is the observed image, it can be corrected to get  $v_c$  according to:

$$v_c(x, y) = \left( v(x, y) - I_{\text{black}}(x, y) \right) / \left( I_{\text{white}}(x, y) - I_{\text{black}}(x, y) \right)$$

## 4.5 CRF Estimation $g(\cdot)$

Once the non-uniform illumination has been compensated, the camera response function can finally be estimated and then the non-linear response of the sensors corrected. The estimation and correction procedure is based on a strong property of the proposed pattern: the noise pattern's region was generated assigning equal probabilities to black and white values (0 and 1 respectively, after normalization). Consequently, since the PSF has unit area, the mean gray value within the observed image should be 0.5.

The solution CRF is defined as a parabolic function  $u \rightarrow \alpha u^2 + (1 - \alpha)u$  where  $\alpha$  is chosen so that the mean of the pattern after the correction is 0.5. Hence,  $\alpha$  can be calculated as:

$$\alpha = \frac{0.5 - \sum u}{\sum (u^2 - u)},$$

where  $\alpha$  is well defined as long as  $u$  is not a binary image which will never happen in our case.

## 4.6 Pattern Rasterization

Rasterization is the procedure of converting a vector image into a raster pixel image. As we have a vector description of the pattern image we can rasterize it at any desired resolution. For that, it must be interpolated at the desired resolution by taking into account the estimated geometrical transformation. Then, we also need to cut the spectrum of the rasterized image to be band-limited at twice the desired frequency at which the PSF is estimated. We do this by the following procedure:

1. The continuous pattern  $u$  is sampled at a very high resolution, e.g. each of the flat black or white squares is sampled with a  $4 \times 4$  block of pixels. We are going to work directly with these samples  $\mathbf{u}$  (digital image), instead of the continuous pattern.
2. Frequencies higher than  $s\pi$  are cut off from the digital pattern  $\mathbf{u}$  to get  $\tilde{\mathbf{u}}$ . The DCT of  $\mathbf{u}$  is computed and the DCT coefficients larger than  $(m \times s, n \times s)$  are set to zero ( $m$  and  $n$  are the number of rows and columns of the noise part in the captured image). Then we recover the filtered version  $\tilde{\mathbf{u}}$  by applying the Inverse DCT. Note that  $\pi$  represents the camera sensor sampling frequency that is why the factors  $m$  and  $n$  are included.
3. Using the previously computed geometric distortion, the filtered pattern  $\tilde{\mathbf{u}}$  is scaled to the desired resolution  $s$  by bicubic interpolation. The resultant image  $\tilde{\mathbf{u}}_D$  is of size  $(s \times (m - 1) + 1, s \times (n - 1) + 1)$ .

## 4.7 Building the Linear System

In order to find the PSF, we generate the following linear system:

$$\operatorname{argmin}_{\mathbf{h}} \quad \|\mathbf{S}_s \mathbf{U} \mathbf{h} - \mathbf{v}\|_2^2, \quad (\text{P})$$

The matrix  $\mathbf{S}_s \mathbf{U}$  is composed by:

1. The matrix  $\mathbf{U}$  associated to the 2D convolution operator with the image  $\tilde{\mathbf{u}}_D$ . The convolution is done with a kernel of size  $p \times q$ .
2. The s-down-sampling matrix  $\mathbf{S}_s$  takes one sample per each block of  $s \times s$  pixels.
3. Finally a mask is applied that puts to zero all values that are outside the *region of interest*.

The *region of interest* consists of a trapezoid mask that restricts the convolution to the noise part of the pattern. Also this mask is eroded by a factor  $(\max(p, q) - 1)/2s$  to avoid boundary problems due to the convolution of finite support sequences.

We rewrite the observed image  $v$  in vector form to be consistent with the matrix formulation of the system. The mask restricting to the region of interest is also applied to the observation  $v$ .

## 4.8 Numerical Methods for PSF Estimation

Finally, we need to solve for a non-negative PSF

$$\operatorname{argmin}_{\mathbf{h}} \quad \|\mathbf{S}_s \mathbf{U} \mathbf{h} - \mathbf{v}\|_2^2, \quad (\text{P})$$

subject to  $\mathbf{h}_i \geq 0$ ,  $i = 1, \dots, r^2$ .

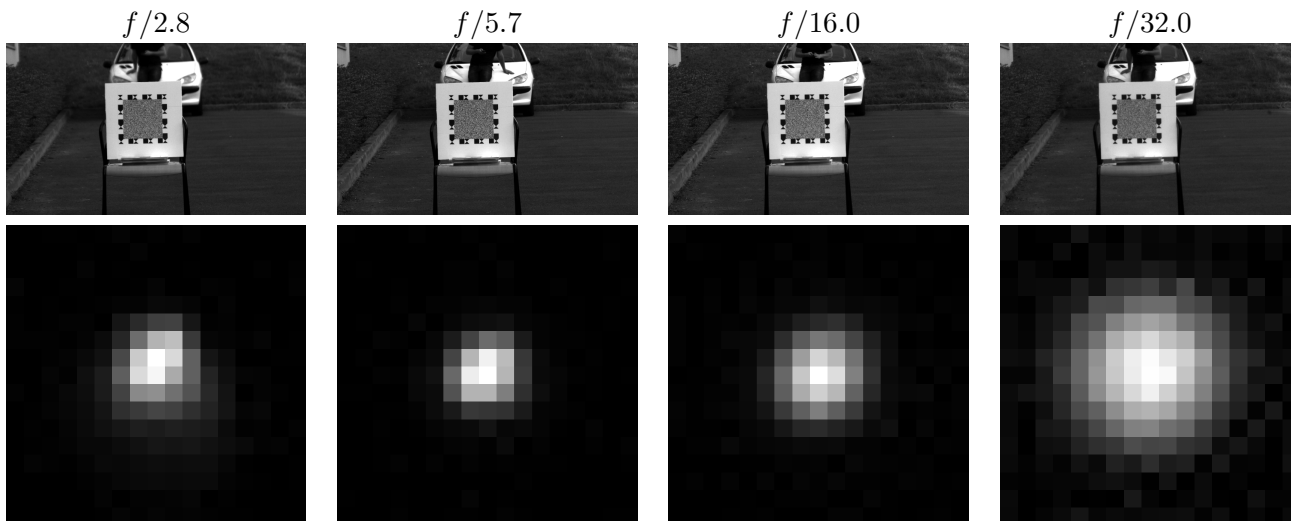
To directly solve the non-negative least squares problem (P) we used the Newton interior point algorithm proposed by Portugal et al. [4] and if we release the non-negative hypothesis the solution is simply found by a least squares algorithm. Another option is to threshold the least squares solution to be non-negative. The attached source code can operate in any of these three options.

## 5 Examples

In this section we present some local  $4 \times$  PSF estimation examples from real camera acquisitions. In all cases a Canon EOS 400D camera provided with a Tamron AF 17-50mm F/2.8 XR Di-II lens was used. The focal length was fixed at 50.0 mm. Based on these experiments the behavior of the proposed method was analyzed with varying camera aperture. Variations of the kernel estimates depending on the least squares numerical algorithm are also explored. The raw output values of all the PSF estimations shown are included as Supplementary Material and can be downloaded from the article web page <https://doi.org/10.5201/ipol.2012.admm-nppsf>.

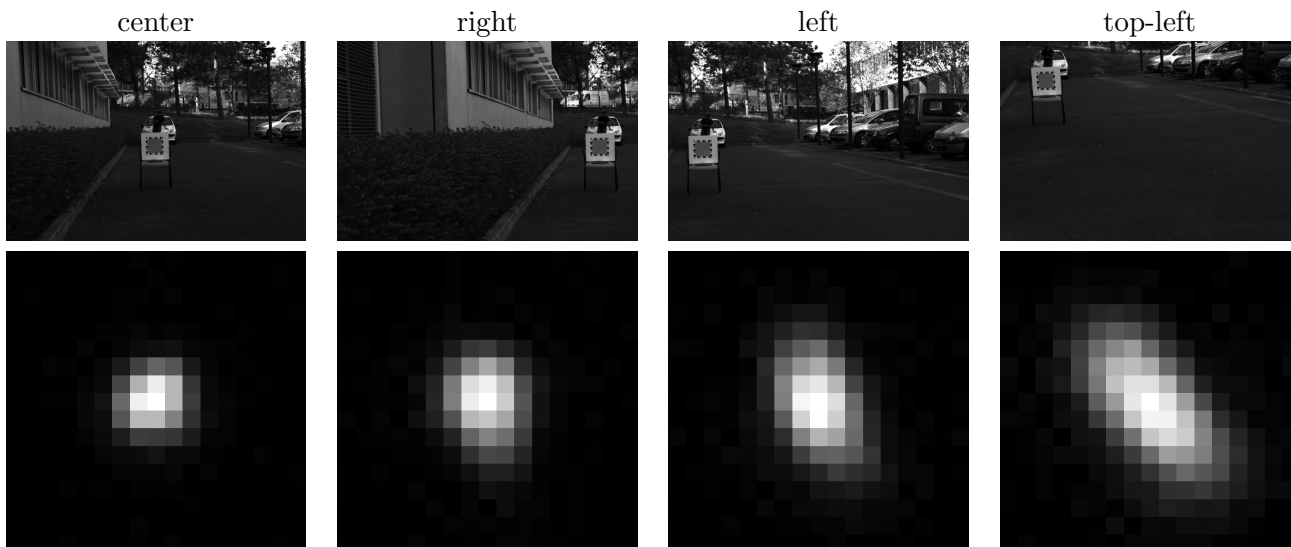
### 5.1 Different Camera Apertures

Here we show kernel estimations for four different apertures. The PSF are estimated at  $4 \times$  for one of the green channels (half of the green pixels of the Bayer matrix). The estimations were performed at the image center and the non-negative least squares algorithm was used. The support was set to  $17 \times 17$  in all cases. Notice that kernels at apertures  $f/32$  and  $f/16$  are significantly larger than the rest, as predicted by diffraction theory. The PSF at aperture  $f/2.8$  is more anisotropic than in the other cases implying some kind of lens aberration only noticeable when the lens is wide open.



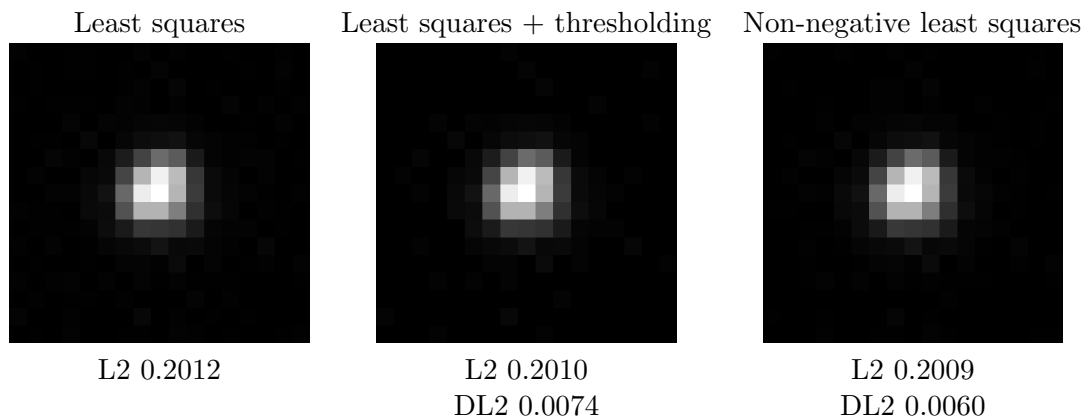
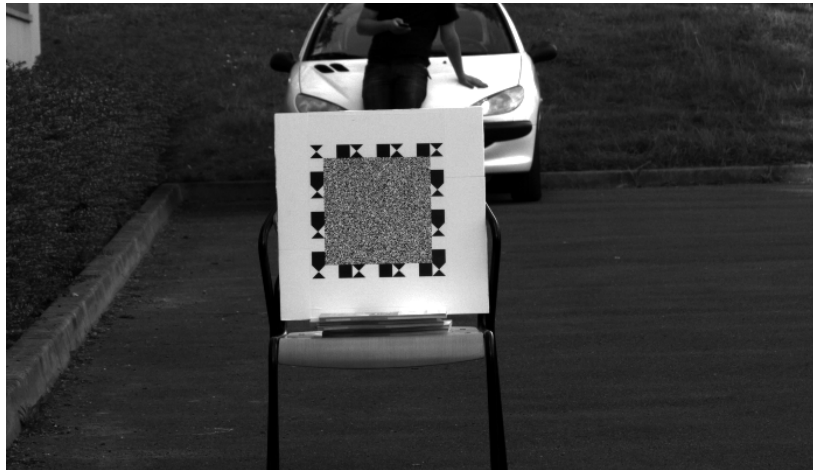
## 5.2 Locations

In this experiment we show the result of estimating a  $4\times$  PSF for one of the green channels, at different image locations, for  $f/5.7$ . Kernels closer to image borders are larger and more asymmetrical than the kernel at the image center. This seems to be a consequence of lens aberrations that deteriorate the system performance.



## 5.3 Least Squares Numerical Algorithm and the Non-negativity Constraint

The following experiment shows the results when using the different variants of the least squares numerical algorithm. The PSFs are estimated at  $4\times$  for one of the green channels and the support was set to  $17\times 17$  in all cases. Notice that all the kernels are very similar. In particular, not enforcing the kernel to be non-negative yields essentially the same results. In fact, this is a sanity check on the proposed method.

Captured image  $f/5.7$  focal 50.0 mm

L2 is the standard  $\ell_2$  euclidean norm of the PSF and DL2 is the  $\ell_2$  norm of the difference between the estimation and the least squares estimation.

## Credits



The vector image of the calibration pattern is licensed under the Creative Commons Attribution License (CC-BY).

## References

- [1] M. Delbracio, P. Musé, A. Almansa and J.M. Morel, *The non-parametric sub-pixel local point spread function estimation is a well posed problem*, International Journal of Computer Vision, vol. 96, no. 2, pp. 175-194, January 2012. <http://dx.doi.org/10.1007/s11263-011-0460-0>.
- [2] R. Grompone, J. Jakubowicz, J.M. Morel and G. Randall, *LSD: a Line Segment Detector*, Image Processing On Line. [http://www.ipol.im/pub/alg/gjmr\\_line\\_segment\\_detector/](http://www.ipol.im/pub/alg/gjmr_line_segment_detector/), preprint, accessed March 16, 2012.
- [3] R. Sprengel, K. Rohr and H.S. Stiehl. *Thin-plate spline approximation for image registration*. Engineering in Medicine and Biology Society, 1996. Bridging Disciplines for Biomedicine. Proceedings of the 18th Annual International Conference of the IEEE, 1190-1191 Vol 3, Year 1996. <http://dx.doi.org/10.1109/IEMBS.1996.652767>.

- [4] L. F. Portugal, J.J. Júdice and L.N. Vicente. *A comparison of block pivoting and interior-point algorithms for linear least squares problems with nonnegative variables*. Mathematics of Computation, Number 208, Pages: 625-643, American Mathematical Society, Volume 63, Year 1994. <http://dx.doi.org/10.2307/2153286>.