École Normale Supérieure de Cachan, France
Universidad de la República, Uruguay

Two Problems of Digital Image Formation:

Recovering the Camera Point Spread Function
and
Boosting Stochastic Renderers by Auto-similarity Filtering

A dissertation presented
by

Mauricio Delbracio Bentancor

in fulfillment of the requirements
for the degree of Doctor of Philosophy
in the subject of Applied Mathematics

Committee in charge

Referees	Sylvain DURAND	-	Université Paris Descartes, FR
	Patrick PÉREZ	-	Technicolor Research, FR
	Guillermo SAPIRO	-	Duke University, USA
Advisors	Jean-Michel MOREL	-	ENS de Cachan, FR
	Pablo MUSÉ	-	Universidad de la República, UY
	Andrés ALMANSA	-	Télécom ParisTech, FR
Examiners	George DRETTAKIS	-	INRIA Sophia-Antipolis, FR
	Saïd LADJAL	-	Télécom ParisTech, FR

March 2013

This dissertation was defended on March 25, 2013.
Version of October 23, 2013 at 17:45.

Abstract of the Dissertation

Mauricio Delbracio Bentancor

Two Problems of Digital Image Formation:
Recovering the Camera Point Spread Function
and
Boosting Stochastic Renderers by Auto-similarity Filtering

Under the direction of:
Jean-Michel Morel, Pablo Musé and Andrés Almansa

This dissertation contributes to two fundamental problems of digital image formation: the modeling and estimation of the blur introduced by an optical digital camera and the fast generation of realistic synthetic images.

The accurate estimation of the camera's intrinsic blur is a longstanding problem in image processing. Recent technological advances have significantly impacted on image quality. Thus improving the accuracy of calibration procedures is imperative to further push this development.

The first part of this thesis presents a mathematical theory that models the physical acquisition of digital cameras. Based on this modeling, two fully automatic algorithms to estimate the intrinsic camera blur are introduced. For the first one, the estimation is performed from a photograph of a specially designed calibration pattern. One of the main contributions of this dissertation is the proof that a pattern with white noise characteristics is near optimal for the estimation purpose. The second algorithm circumvents the tedious process of using a calibration pattern. Indeed, we prove that two photographs of a textured planar scene, taken at two different distances with the same camera configuration, are enough to produce an accurate estimation.

In the second part of this thesis, we propose an algorithm to accelerate realistic image synthesis. Several hours or even days may be necessary to produce high-quality images. In a typical renderer, image pixels are formed by averaging the contribution of stochastic rays cast from a virtual camera. The simple yet powerful acceleration principle consists of detecting similar pixels by comparing their ray histograms and letting them share their rays. Results show a significant acceleration while preserving image quality.

Résumé de la Thèse

Mauricio Delbracio Bentancor

Deux problèmes dans la formation des images numériques :
l'estimation du noyau local de flou d'une caméra
et
l'accélération de rendus stochastiques par filtrage auto-similaire

Sous la direction de :
Jean-Michel Morel, Pablo Musé and Andrés Almansa

Cette thèse s'attaque à deux problèmes fondamentaux dans la formation des images numériques : la modélisation et l'estimation du flou introduit par une caméra numérique optique, et la génération rapide des images de synthèse photoréalistes.

L'évaluation précise du flou intrinsèque d'une caméra est un problème récurrent en traitement d'image. Des progrès technologiques récents ont eu un impact significatif sur la qualité de l'image. Donc, une amélioration de la précision des procédures de calibration est impérative pour pousser plus loin cette évolution.

La première partie de cette thèse présente une théorie mathématique de l'acquisition physique de l'image par un appareil photo numérique. Sur la base de cette modélisation, deux algorithmes automatiques pour estimer le flou intrinsèque de la l'appareil sont proposés. Pour le premier, l'estimation est effectuée à partir d'une photographie d'une mire d'étallonnage spécialement conçue à cet effet. L'une des principales contributions de cette thèse est la preuve qu'une mire portant l'image d'un bruit blanc est proche de l'optimum pour estimer le noyau de flou. Le deuxième algorithme évite l'utilisation d'une mire d'étallonnage, procédure qui peut devenir un peu encombrante. En effet, nous montrons que deux photos d'une scène plane texturée, prises à deux distances différentes avec la même configuration de l'appareil photo, suffisent pour produire une estimation précise.

Dans la deuxième partie de cette thèse, nous proposons un algorithme pour accélérer la synthèse d'images réalistes. Plusieurs heures, et même plusieurs jours peuvent être nécessaires pour produire des images de haute qualité. Dans un rendu typique, les pixels d'une image sont formés en établissant la moyenne de la contribution des rayons stochastiques lancés à partir d'une caméra virtuelle. Le principe d'accélération, simple mais puissant, consiste à détecter les pixels similaires en comparant leurs histogrammes de rayons et à leur faire partager leurs rayons. Les résultats montrent une accélération significative qui préserve la qualité de l'image.

Resumen de la Tesis

Mauricio Delbracio Bentancor

Dos problemas en la formación de imágenes digitales:
la estimación de la función de dispersión de punto de una cámara fotográfica
y
la aceleración de renderers estocásticos por filtrado auto-similar

Bajo la dirección de:
Jean-Michel Morel, Pablo Musé and Andrés Almansa

Esta tesis contribuye a resolver dos problemas fundamentales en la formación de imágenes digitales: (i) el modelado matemático y la estimación de la falta de definición introducida por el sistema óptico/electrónico de una cámara digital y (ii) la generación rápida de imágenes sintéticas fotorealistas.

La estimación precisa del núcleo de convolución (por falta de definición) intrínseco a la cámara es un problema importante en procesamiento de imágenes. Los avances tecnológicos recientes han impactado significativamente en la calidad de las imágenes, por lo que una mejora en la exactitud de los procedimientos de calibración resulta imprescindible para impulsar aún más este desarrollo.

La primera parte de esta tesis presenta una teoría matemática que modela la adquisición física de una imagen por una cámara digital. Sobre la base de este modelo, presentamos dos algoritmos totalmente automáticos para estimar la falta de definición intrínseca de la cámara. En el primero, la estimación se realiza a partir de una fotografía de un patrón de calibración que contiene un ruido blanco especialmente diseñado. La prueba de casi-optimalidad de dicho patrón, en el sentido del condicionamiento numérico del problema de estimación, constituye una de las principales contribuciones de esta tesis. El segundo algoritmo simplifica el procedimiento experimental al no requerir el uso de un patrón de calibración predeterminado. De hecho, se prueba que dos fotografías de una escena plana texturada, tomadas a dos distancias diferentes con la misma configuración de la cámara, son suficientes para producir una estimación precisa.

En la segunda parte de esta tesis, se propone un algoritmo para acelerar la síntesis de imágenes fotorealistas. Para producir imágenes de alta calidad pueden ser necesarias varias horas o incluso días. En un motor típico de renderizado, los píxeles de una imagen se forman haciendo un promedio de la contribución de rayos emitidos aleatoriamente desde una cámara virtual. El principio de aceleración propuesto, simple pero poderoso, consiste en detectar píxeles similares mediante la comparación de sus histogramas de rayos, y hacer que compartan sus rayos. Los resultados muestran que es posible obtener una aceleración considerable preservando la calidad de imagen.

A mis viejos Carlos y Olga.

Acknowledgments

First and foremost I would like to express my heartfelt gratitude to my advisors Andrés Almansa, Jean-Michel Morel and Pablo Musé, with whom I have had the tremendous privilege to work for the past four years. The enormous willingness of Andrés to discuss ideas, his creativity and scientific rigor have profoundly marked this thesis. His unfailing good mood and constant optimism have made this work very enjoyable. It is hard to imagine a more generous advisor than Jean-Michel, from whom I not only have learned a lot of technical insights but also a research methodology based on mathematical thinking. His wise advice during our frequent meetings allowed me to work with total freedom in a very comfortable environment. My dear friend Pablo, who guided me and continuously transmitted his personal experience and humanism over these years. His continuous and total support along with his scientific advice and knowledge allowed me to always have confidence in our work.

A special thanks goes to Sylvain Durand, Patrick Pérez, Saïd Ladjal, Guillermo Sapiro and George Drettakis, who generously accepted being part of the evaluation committee and provided me with encouraging and constructive feedback.

I am grateful to Rafael Grompone von Gioi who, from the time I stepped on french soil, was there to help me with a lot of practical problems. Rafael, from whom I have learned a lot over the many discussions we have had over these years.

I want to warmly thank Tony Buades. This work has benefited from his remarkable collaboration. I would also like to thank Nicholas Phelps and Julien Chauvier from E-On Software who introduced me in a passionate domain.

I am grateful to all the people that helped me during these years, in particular I would like to thank: Alejandro, Chloé, Eduardo, El Gordo, Fede Larroquet, Ives, Jairo, Jimena, José, Juliana, Muhsin, Natalia, Natalie, Paola, Rose and Sambita, with whom I have shared some moments and many beers.

This thesis has been part of a very close relationship between the Electrical Engineering Department (IIE) of the Engineering Faculty at the Universidad de la República in Montevideo and the Mathematical Department (CMLA) of the ENS-Cachan. Working between the CMLA and the IIE has been a tremendous pleasure, thank you specially to Véronique Almadovar, Micheline Brunetti, Sandra Doucet, Virginia Pauchont, Carine Saint Prix for their efficiency and patience with my limited French language skills; and to Laura Landin and Maria Misa who always helped me (and very efficiently!) with all my paperwork.

During these years I have had the privilege of working with several colleagues in France and Uruguay and I thank them all deeply. Many thanks to Gabriele Facciolo, with whom we dived into the deep waters of the functional analysis.

I want to thank Gregory Randall, who with his momentum and energy formed an exceptional group of researchers working in image and signal processing in Uruguay. I would also like to thank Alicia Fernandez, who recruited me several years ago to work with her and another great friend, German Capdehourat, in what was my first formal research project.

I do not forget my friends, who were always present even though I could not see them very often during these years.

I want to thank from the bottom of my heart my mother, my father and my sister to whom I owe the most. It is only through their efforts, their confidence and their constant encouragement that this thesis has been successful.

Being kilometers away from one's homeland is a tremendous experience. During these years I have realized how lucky I am to have the wonderful company of Cecilia, with whom I share this adventure of crossing the ocean to face new experiences. She has always had infinite patience and gave me her total and unconditional support. Gracias.

Contents

1 Introduction **15**

I Recovering the Camera Point Spread Function 29

2 PSF Estimation from a Calibration Pattern Image **31**
2.1 Introduction 31
2.2 Image Formation Model 35
2.3 Optimality Criterion and Quality Measure for Calibration Patterns 37
2.4 The Complete PSF Estimation Procedure 47
2.5 Experimental Results 51
2.6 Discussion 62

3 PSF Estimation from Two Photographs at Different Distances **65**
3.1 Introduction 65
3.2 Image Formation Model 67
3.3 PSF Estimation from an Unknown Pair of Scaled Images 70
3.4 The Complete PSF Estimation Procedure 75
3.5 Experimental Results 76
3.6 Conclusion 82
3.A Mathematical Framework and Physical Modeling 85
3.B Proof of Auxiliary Results 88

II Accelerating realistic image synthesis 91

4 Boosting Monte Carlo Renderers **93**
4.1 Introduction 93
4.2 Previous Work 95
4.3 Proposed Approach 97
4.4 Implementation Details 103
4.5 Experimental Set-up and Results 104
4.6 Discussion, Limitations and Future Work 126
4.7 Conclusion 126

Conclusions and Perspectives **131**

A Detailed Description of the PSF Estimation Algorithms **135**
A.1 PSF Estimation from a Calibration Pattern Image 135
A.2 PSF estimation from Two Photographs at Different Distances 140

Bibliography **143**

1 Introduction

Motivation

Digital images are generated by using physical acquisition devices, such as digital cameras, but also by simulating light propagation through environmental models. This thesis deals with two fundamental problems of digital image generation: the modeling and estimation of the camera's intrinsic blur presented in all optical digital devices and the generation of fast and realistic synthetic images.

Mostly due to advances in technology, pushed by a massive market penetration, digital image quality has significantly improved in the past two decades. This has caused that certain characteristic problems, caused by the nature of the acquisition devices, that were not critical until recently, are now becoming the bottleneck to get further improvements. Blur produced by light diffraction, geometrical distortions caused by the use of low-cost lenses and thermal noise due to electronic circuits are examples of this kind of outcomes.

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). In addition to 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. The first part of this dissertation, addresses the problem of accurately estimating 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.

At the other end, synthesizing high quality realistic images in a reasonable amount of time remains a major challenge in computer graphics. The aim of realistic image synthesis is to generate new images from a complete three-dimensional description of a virtual scene. The scene description should contain at least the geometry, location and properties of objects, the camera viewpoint and a characterization of light sources. The generated picture should be as photorealistic as possible: if the three-dimensional scene is constructed and a photograph is taken from the same camera's point of view, the difference should be negligible. Of course this requires a perfect knowledge of how the light interacts with the environment and extremely accurate material models; oversimplifications must be avoided.

The seminal paper by Kajiya [1986] presented the *rendering equation*, an integral equation modeling the steady-state light distribution in a scene. Except for very simple scenes, analytical solutions are impossible to obtain, so most typical approaches are based on Monte Carlo numerical integration techniques. Image pixels are formed by averaging the contribution of stochastic rays cast from a virtual camera through the scene. The principal problem of Monte Carlo renders is that the variance of the estimator decreases linearly with the number of stochastic samples. Thus the root mean square error to an ideal image decreases as the square root of the number of samples. Several hours or even days may be necessary to produce noiseless realistic images. Indeed, at present, the final image quality is indirectly topped by the available production time and computational resources.

This makes this problem interesting not only from an academic point of view. Indeed, the motivation for this problem came through a partnership with the French company *e-on software*[1], specialist in modeling and rendering natural environments. The company's CEO, Nicholas Phelps proposed a concrete well defined problem:

> Is there any way of generating realistic synthetic images, more quickly, with the same amount of computational resources and without loosing quality?

The second part of this dissertation analyzes some ideas and proposes a new method to accelerate Monte Carlo renderers, which are the most popular realistic renderers that are currently used.

Part I – Recovering the Camera Point Spread Function

The point spread function (PSF) describes the distribution of light in the camera focal plane for a point light source. This function is strictly related to the resolution and blur of an optical device. Most medium to high quality digital cameras (DSLRS) acquire images at a spatial rate which is below the Nyquist rate. For this reason only aliased versions of the camera point-spread function can be directly observed. In addition, since the acquisition system is only locally stationary, the PSF estimation must be local.

PSF estimation methods can be classified as *blind* or *non-blind*, depending on whether they use or not snapshots of a specially designed calibration pattern. Blind approaches try to estimate the PSF from photographs of an unknown scene. They do assume, however, that the scene involved in the estimation follows some statistical model of sharp images, or includes a significant amount of geometric cues such as sharp edges. Most of these PSF estimation approaches attempt to detect edges, which are modeled as pure step-edge functions convolved with the PSF kernel [Chalmond 1991; Luxen and Förstner 2002; Capel 2004; Smith 2006]. In this setting, the 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. Therefore, such blind estimation techniques do not normally lead to accurate PSF estimates and are only used in image restoration problems, where precision is not the main objective. For this reason, most accurate PSF estimation procedures rely on the use of specially designed calibration patterns. A local kernel estimation is performed by comparing the ideal calibration pattern to its photographs.

In this thesis, two fully automatic algorithms for the PSF estimation are introduced. The first one performs an estimation from a photograph of a specially designed calibration pattern. As we will show, this non-blind algorithm achieves high accuracy. The second one uses two photographs of a planar scene, taken at two different distances with the same camera configuration. This algorithm lays in an intermediate category: *semi-blind*. Although, it does not make use of a PSF estimation pattern, the photographed scene should be planar and textured to get accurate results.

---

[1] `http://www.e-onsoftware.com/`

Pursuit of the Optimal Calibration Pattern

Up to the present, even non-blind subpixel PSF estimation methods reported in the literature led to ill-posed inverse problems. The inversion required the imposition of simple PSF parametric models, or other regularity or symmetry priors. In Chapter 2 we show that such *a priori* assumptions on the PSF are actually unnecessary and jeopardize the estimation accuracy.

A mathematical digital image formation model that takes into account blur due to intrinsic and extrinsic phenomena, geometrical lens distortions, sampling and noise is presented in Section 2.2. One of the main problems for the PSF estimation, is that digital cameras may capture images that are undersampled according to Shannon theory. The mathematical model, built on Lemma 1, considers this issue specifically. In this discrete model, images are sampled in a virtual super-resolved grid. This high-resolution lattice will be for example, 4×4 times the resolution of the physical camera allowing a $4\times$ PSF estimation.

Assume we can unveil exactly the latent sharp pattern image that produced the blurry aliased observation. Then, solving for the PSF amounts to solve an inverse problem governed by the image formation model. To achieve this several problems need to be addressed:

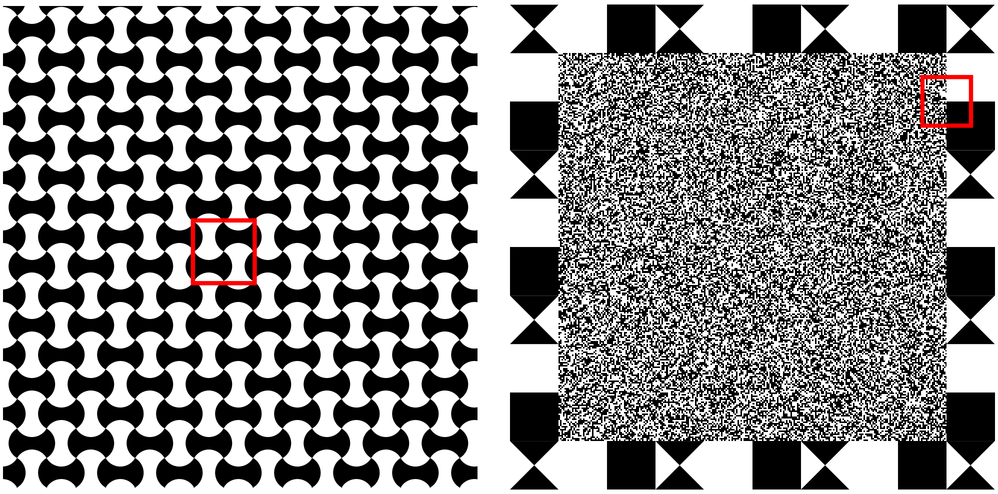
- how to choose a good PSF calibration pattern;
- how to estimate the geometric deformation between the pattern and the acquired image;
- how to estimate the non-uniform illumination;
- how to numerically solve the inverse problem.

If the pattern and the captured image are perfectly registered and its non-uniform illumination compensated, the accuracy of the PSF estimation depends on how well we can invert an operator. This operator mainly depends on the calibration pattern. In Section 2.3 we present a quality measure (that we call the γ value) of a given pattern view in terms of the well-posedness of the PSF estimation problem. Formalizing this well-posedness measure allows us to define an *optimal digital pattern* for the subpixel PSF estimation (Definition 1).

Is it feasible to construct such an ideal calibration pattern? Several patterns have been used for PSF estimation, ranging from pin-hole, slanted-edge [ISO 2000; Reichenbach et al. 1991; Zandhuis et al. 1997; Claxton and Staunton 2008], or arc-step-edge patterns [Joshi et al. 2008] to random noise images [Daniels et al. 1995; Levy et al. 1999; Backman et al. 2003, 2004; Brauers et al. 2010].

A theoretical bound on the optimality of a pattern is presented in Proposition 1. This mathematical bound gives a limit of the performance that we can theoretically achieve. Several numerical experiments conducted in Section 2.3 allow us to conclude that in realistic conditions, near-optimal quality measure values are reached with a pattern created from a realization of a random field of independent Bernoulli black or white pixels.. The mathematical modeling of the PSF estimation problem together with the near-optimality of the Bernoulli pattern is one of the main contributions of this dissertation.

Figure 1.1 shows the proposed Bernoulli noise pattern, compared to the pattern designed by Joshi et al. [2008] consisting of 120° arc step edges. The proposed noise pattern consists of 256×256 small black/white squares drawn independently from a Bernoulli equiprobable distribution. This central region is surrounded by checkerboard-like marks and black/white squares for alignment and illumination estimation purposes.The fact that the pattern is black and white avoids to calibrate the printer used for generating the pattern.



Slant-edge pattern by Joshi et al. [2008]

Bernoulli pattern (proposed)

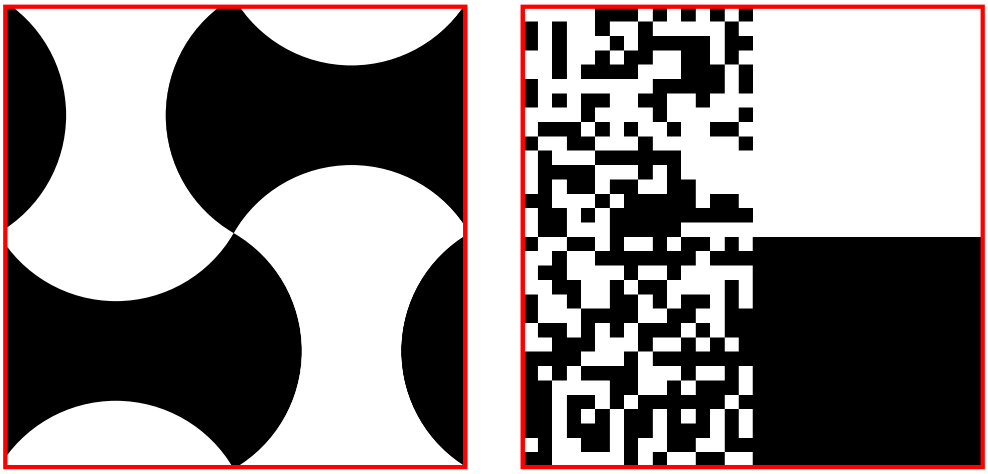


Figure 1.1: Different calibration patterns for local PSF estimation. On left the Joshi et al. pattern consisting of 120° arc step edges. On the right the proposed noise pattern consists of 256×256 small black/white squares drawn independently from a Bernoulli equiprobable distribution. The lateral checkerboard-like marks and black/white squares are introduced for alignment and illumination estimation purposes.

	9×9	17×17	25×25	33×33
Joshi et al. [2008]	99.44	1133.05	6445.87	58419.08
Bernoulli pattern	0.19	0.69	1.54	2.98
Theoretical bound	0.10	0.35	0.70	1.15

Table 1.1: Pattern quality measure. The value shown in each entry (γ value) is a measure of the well-posedness of the subpixel PSF estimation problem. The larger the γ value the more ill-posed the problem becomes. The Bernoulli pattern produces significantly smaller γ values than the slanted-edge Joshi et al. [2008] pattern. Values are calculated for a $4\times$ PSF estimation for PSF support sizes ranging from 9×9 to 33×33.

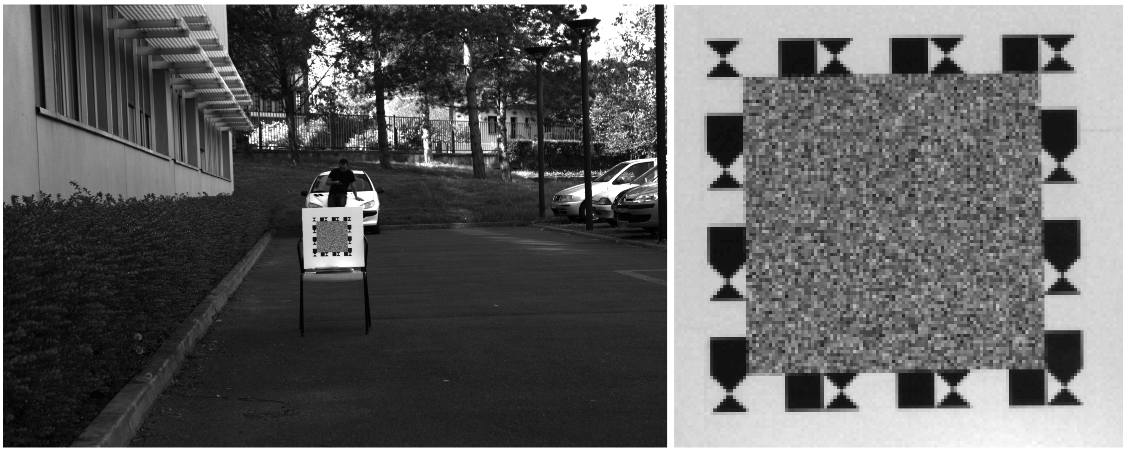


Figure 1.2: Example of an acquired image of the Bernoulli pattern. On the right, a crop of the central part containing the pattern. As is shown on the left, the pattern should cover only a small region of the image (roughly 100×100 pixels). This allows for a local PSF estimation.

The well-posedness measure for the corresponding patterns and the theoretical optimal values are shown in Table 1.1. The Bernoulli noise pattern significantly outperforms the one from Joshi et al. [2008].

The pattern is designed to cover a small region of 100×100 pixels in the acquired image. This permits an estimation which is both local and well-posed. An example of an acquired image is shown in Figure 1.2. This image shows one of the green channels of the Bayer raw camera output.

The proposed non-blind PSF estimation algorithm is fully automatic. The captured image is precisely aligned to the analytic pattern by means of the surrounding checkerboard markers. Non-uniform illumination is corrected from the acquired image by using the auxiliary black and white flat regions. Non-linear sensor response is also roughly estimated based on the fact that the central noise part of the pattern should have a perfect average of black and white. Of course, since we work with raw camera output, this response is almost linear.

Once these intermediate steps are performed, the local PSF is directly computed by inverting a linear system. Since the PSF must be non-negative, as we justify in the image formation model, a numerical constraint is imposed to enforce this non-negativity. Notwithstanding, the experimental section shows that not enforcing the kernel to be non-negative essentially yields the same results. In fact, this serves as a sanity check on the proposed method.

In order to validate the calibration procedure and the quality of the Bernoulli pattern we conducted several synthetic and real camera experiments. A comparison to the commercial software

Imatest [LLC 2010] and the state-of-the-art method of Joshi et al. is presented in Figure 1.3. This figure shows the horizontal profile of the modulus of the PSF Fourier spectrum. The Imatest estimation is performed from a slanted-edge image and only gives an estimate of the point spread function at the direction orthogonal to the slanted-edge. The Joshi et al. method is forced to use a penalty term on the norm of the PSF gradient, since the inverse problem using their slant-edge pattern is ill-posed. In the low frequencies Joshi et al. and the proposed method yield very similar results. However, for higher frequencies the result of Joshi et al. is strongly dependent on the regularization level. The Imatest estimate is quite noisy and does not resolve frequencies above twice the sampling rate. The proposed algorithm based on the Bernoulli pattern generates much more information than the typical slanted-edge PSF calibration.

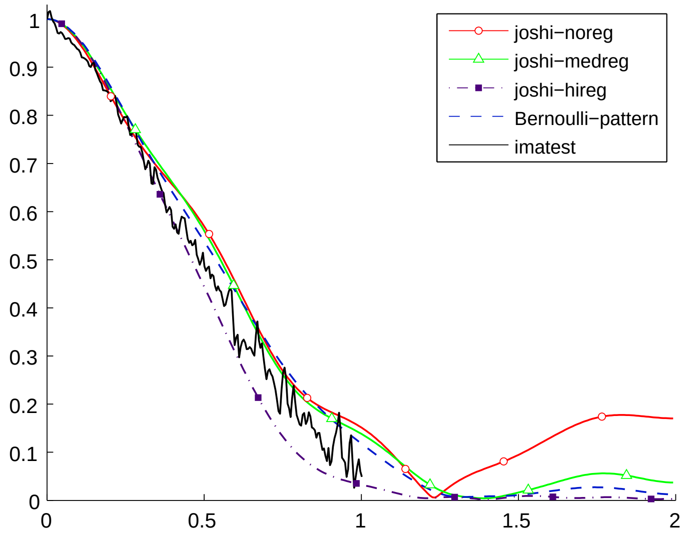


Figure 1.3: Horizontal profile of the PSF Fourier spectrum modulus for different state-of-the-art methods: Joshi et al. [2008] , Imatest [LLC 2010] commercial software and the proposed Bernoulli pattern algorithm. On the low frequencies all algorithms gave very similar estimations, while on the higher frequencies the Joshi et al. estimation depends on the regularization level. The Imatest software produces a quite noisy estimation.

Figure 1.4 shows the result of estimating a 4× PSF for one of the green channels at different image locations using the proposed Bernoulli pattern. Kernels closer to image borders are larger and more asymmetrical than the kernel at the image center. This agrees with the expected result, since lenses are designed to minimize aberrations near the optical center. Note that although no regularization is imposed, the resulting PSFs are smooth.

Avoiding the Use of a Calibration Pattern

Although very precise, the use of a calibration pattern can be sometimes tedious and impractical: these approaches rely on a careful setup, and the calibration grid has to be properly assembled, whereby a good quality print is essential. Therefore, we explored the feasibility of obtaining accurate PSF estimates, while avoiding the explicit use of a calibration pattern.

Chapter 3 proves that, instead of using a photograph of a known calibration pattern, two photographs of the same scene acquired at different distances with fixed camera configuration are

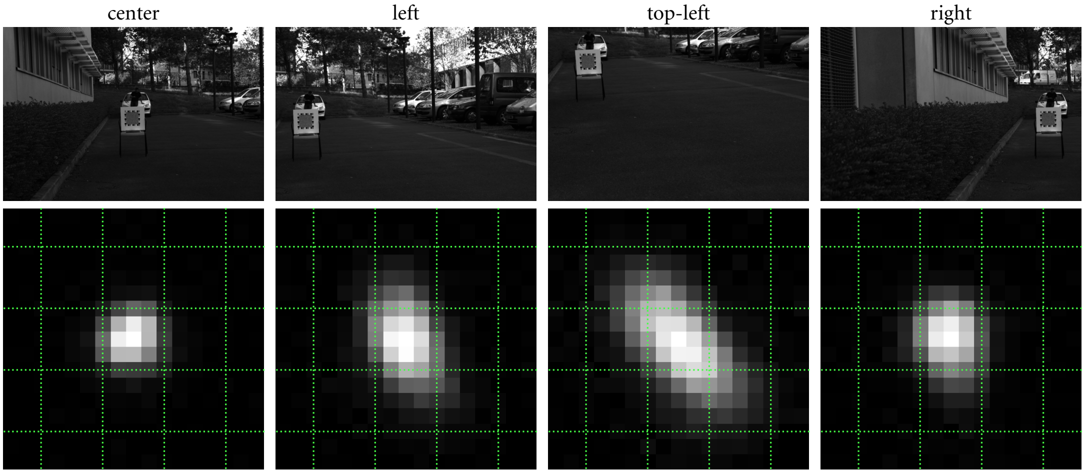


Figure 1.4: The result of estimating a $4\times$ PSF for one of the green channels at different image locations. For each pixel at the sensor resolution we estimate 4×4 samples of the PSF. The $1\times$ resolution (camera grid) is shown in green dotted lines. Kernels closer to image borders are larger and more asymmetrical than the kernel at the image center. This is mainly due to lens aberration. Although no regularization is imposed, the resulting kernels are smooth.

enough to recover a regularization-free subpixel PSF. The mathematical relation between these two fronto-parallel images allows us to introduce the concept of *inter-image* kernel between them. This kernel can be interpreted as the blur that should be applied to the closest image (followed by the necessary zoom-out) to produce the farthest image (Definition 2). An interesting observation, which is proved in Lemma 4, is that the *inter-image* kernel k, and the camera PSF h are closely related through the following equation

$$H_\lambda h * k = h,$$

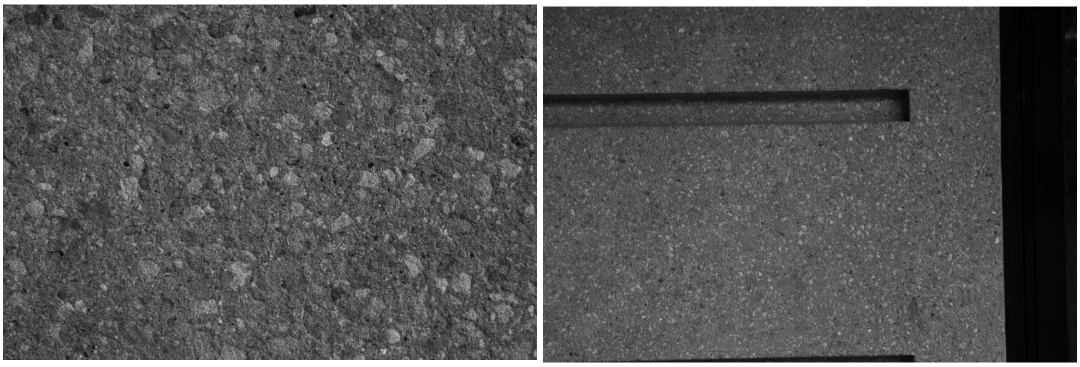
where H_λ is a zoom of the necessary factor to put the two views in the same scale. The derivation of the camera PSF from the previous equation is not straightforward. However, as proved in Proposition 4,

$$h = \lim_{n\to\infty} H_{\lambda^{n-1}} k * H_{\lambda^{n-2}} k * \cdots * H_\lambda k * k.$$

Thus, it is possible to recover the camera PSF from the inter-image kernel k. One of the main contributions of this dissertation is the mathematical development that proves that such an estimation is possible.

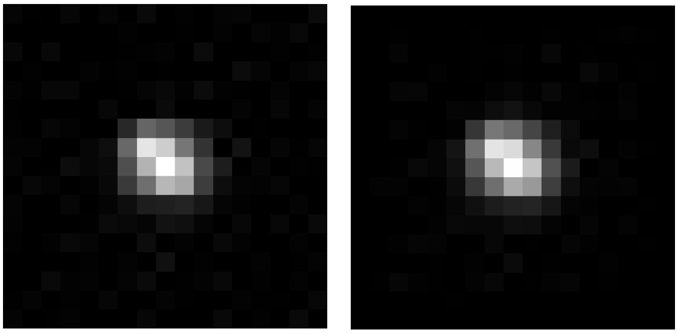
In practice, we deal with discrete images, that may suffer from noise and aliasing, so the inter-image kernel estimation may be biased. Based on a mathematical analysis, we give some ideas on how to mitigate the impact of these problems and to increase the accuracy of the inter-image kernel estimation. This is done by properly choosing the scene and the distance between the acquired images. Indeed, the inter-image kernel is estimated by solving a least squares problem similar to the one for the pattern based PSF estimation. As shown in Chapter 2 the inverse problem is well-posed as long as the photographed scene presents textured characteristics (similar to white noise). In this setting, the closest image plays a similar role as the one played by the calibration pattern in a traditional non-blind estimation.

We introduce an algorithm that is completely automatic. It gives a subpixel estimate of the camera PSF from two photographs of a textured planar object taken at different distances. An



closest image

farthest image



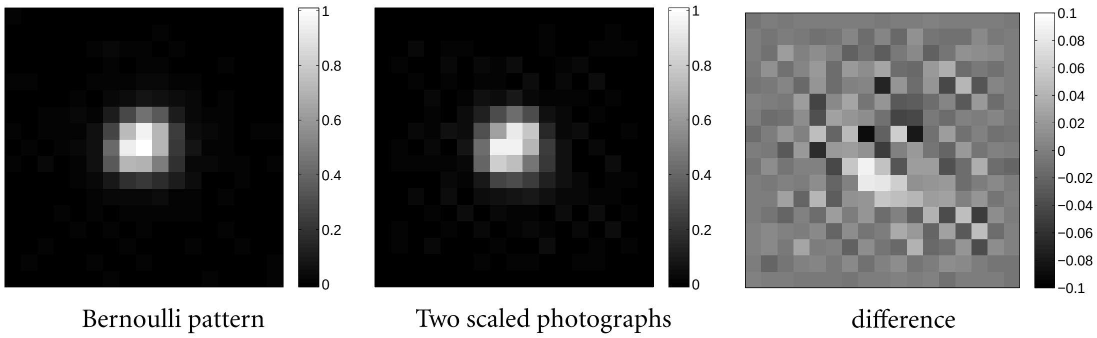
4× Inter-image kernel

4× PSF

Figure 1.5: 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 camera PSF is recovered from the inter-image kernel through an iterative procedure. The estimated inter-image kernel and camera PSF are obtained at $4\times$ the camera resolution for the blue channel. Although no regularization is imposed, the kernels are smooth.

example of how the estimation takes place is shown in Figure 1.5 where a pair of photographs of a wall taken at two different distances are used as input. The algorithm first registers the two input images based on detecting SIFT points [Lowe 1999]. Then, a least-squares problem is solved to get the inter-image kernel estimation. Finally, the camera PSF is computed from the inter image-kernel through an iterative procedure. Experimental evidence shows the well-posedness of the problem and the convergence of the proposed algorithm to the camera in-focus PSF. The mathematical development is made under a technical assumption (e.g., re-focusing does not change the PSF) that may not strictly hold in practice. Nevertheless, as shown in Figure 1.6 the PSF estimated with the Bernoulli pattern and the estimation with the two scaled photographs are convincingly close.

The proposed acquisition procedure is simple and handy in comparison to a non-blind approach. The choice of the photographed scene is important but not critical. For a wide range of everyday textured scenes, the acquired image pairs lead to well posed inversions and accurate results. Figure 1.7 gives an idea of the sharpness of the method. In this experiment, the PSF at four times the camera resolution of the four color Bayer channels (typical color filter array in RAW camera output consists of two green channels, one blue and one red) are estimated. Notice that the red channel PSF is wider than the green and the blue one, as expected from the physics of diffraction-limited optical systems, since the wavelengths associated to red light are larger than the rest. The differences between the dominant orientations of the red/blue and green PSF spectra can be explained by the sensor shape and layout. In fact, each sensor active zone is usually L-shaped (see for example Yadid-Pecht [2000]), and the red and blue sensors are rotated 90° w.r.t. the green ones . These rotations are consistently observed in the PSFs and Fourier spectra estimated with the proposed two photographs method. This illustrates the precision of the proposed approach.



Bernoulli pattern Two scaled photographs difference

Figure 1.6: Comparison of the PSF estimation from the two-scaled photographs and the non-blind estimation using the Bernoulli pattern. The estimations are for the same camera and configuration at $4\times$ the resolution. Both estimations are significantly close and the difference is mostly due to noise present in the two-scale estimation.

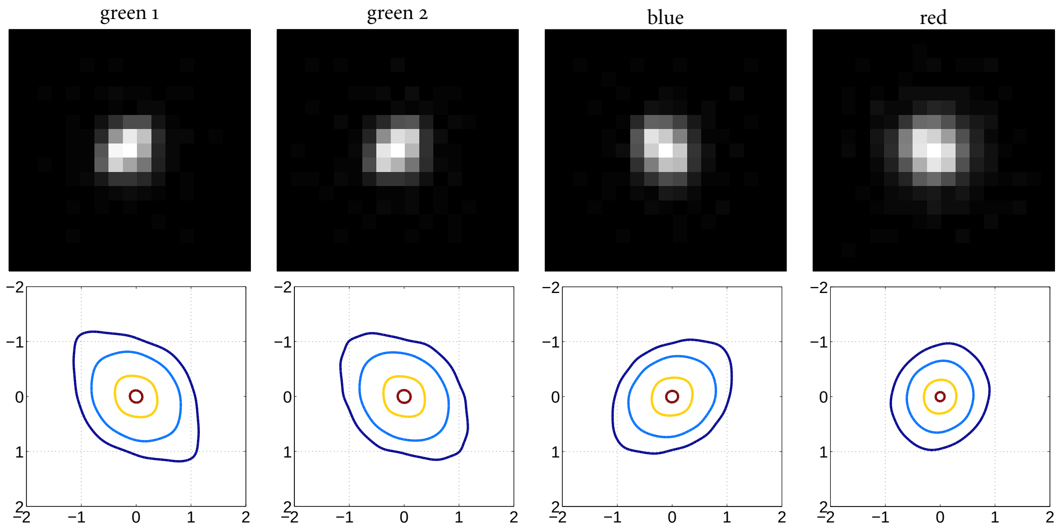


Figure 1.7: Top row: the $4\times$ PSF estimated for the four Bayer channels (two green channels, one red and one blue). Bottom row: their corresponding Fourier spectrum modulus. The red PSF is larger than the blue and the green ones. This is consistent with the diffraction phenomenon: the red wavelengths are larger than the rest, thus its diffraction kernel is wider. Also note the differences between the shape of the red/blue and green PSF spectra (bottom row). Red and blue MTFs are rotated $90°$ with respect to the green ones. This symmetric behavior is consistent with the layout of L-shaped sensors Yadid-Pecht [2000].

Part II – Accelerating Realistic Image Synthesis

In order to synthesize an image with global illumination, a radiance value must be assigned to each pixel in the image. Path tracing (and more generally ray-tracing) is a popular technique for resolving the rendering equation ruling the steady state equilibrium of light in a scene. In a ray-tracing scenario, this value is computed as a weighted average of radiance values incident on the image plane, along light rays coming from the light sources, bouncing in the scene, passing trough the pixel, and pointing to the virtual camera.

Unfortunately, only a finite number of rays can be cast, so the radiance value is computed only approximately. To avoid artifacts, rays are cast randomly. Mathematically, this is equivalent to solving the rendering equation through a Monte Carlo numerical integration procedure. The main problem of Monte Carlo rendering is that the variance of the estimator converges only linearly with the number of random samples. An example of an scene rendered with a varying number of rays per pixel is shown in Figure 1.8.

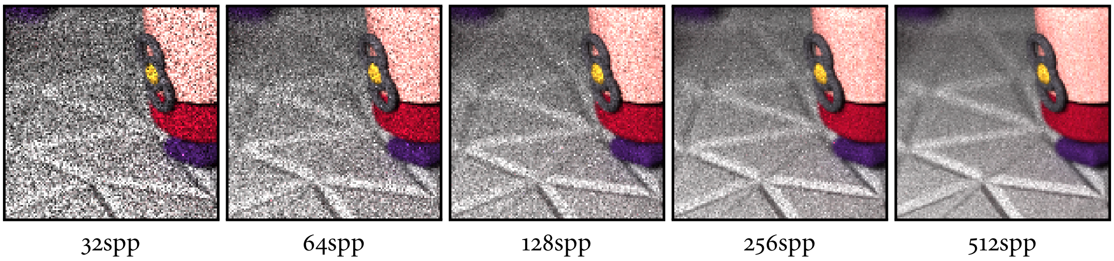


Figure 1.8: Example of an image rendered with Monte Carlo path-tracing. In a pure MC scenario the square error decreases linearly with the number of samples per pixel (spp), thus the convergence is quite slow.

There are mainly two approaches to accelerate the convergence of Monte Carlo rendering to obtain good quality images. One of these approaches is *adaptive sampling*. This class of algorithms locally adapt the number of samples cast per pixel. The idea is to increase the number of rays in complex parts of the scene while maintaining a reduced number in simple parts, such as flat regions. Complex textures or defocused zones are typical elements that require large amounts of rays to be properly rendered. Hachisuka et al. [2008] proposed to adaptively distribute a set of samples in the full, multidimensional sampling domain where the rendering equation is computed. However, as more Monte Carlo effects are considered (e.g. depth-of-field, motion blur, area lighting) the dimension of this space will be larger and thus will suffer from the curse of dimensionality. One of the most significant adaptive sampling algorithms is certainly the Adaptive Wavelet Rendering by Overbeck et al. [2009]. This method adaptively distributes Monte Carlo samples in the screen space to reduce the variance of a wavelet basis scale coefficients. Then, the image is reconstructed from these non-uniformly distributed samples by using a suitable wavelet approximation.

The other approach is *adaptive filtering*. In this family of algorithms, the existing set of samples are combined to produce a better estimator of the pixel color using ray information in a pixel and in its neighbors. Adaptive filtering may take place at sample level (i.e., primarily filtering the ray colors) or at pixel level (i.e., primarily filtering pixel color values). The simplest adaptive filters act at pixel level, like any filter used in classical image processing [Jensen and Christensen 1995; Choudhury and Tumblin 2003; Xu and Pattanaik 2005]. More complex filters make use of ray information available from the renderer in order to filter also at pixel level [Rushmeier and Ward 1994; McCool 1999; Dammertz et al. 2010; Xu et al. 2011]. The most sophisticated filters, use the additional ray information to adaptively filter the sample rays [Shirley et al. 2011; Sen and Darabi 2012; Rousselle et al. 2011; Lehtinen et al. 2012].

The majority of these methods can actually be written as generalized versions of the bilateral filter (or the sigma-filter [Lee 1983]) applying a weighted average of the samples (resp. of the pixels) in a neighborhood. This general bilateral filter obeys the *law of joint destiny (Gesetz des gemeinsamen Schicksals)* introduced in Gestalt psychology by Wertheimer [1923]. This law states that similar pixels are grouped by our perception. Its obvious generalization in image processing is to say that similar pixels must be denoised jointly, being different samples of the same model. The law of common destiny is implicitly used by the sigma-filter and by the NL-means algorithm [Buades et al. 2005]. The main disadvantage of traditional bilateral filters is that by comparing noisy pixel values, they cannot easily distinguish noise from intrinsic pixel variability. Thus, the clustering of similar pixels is potentially subject to errors and the filtering will result in a significantly biased image.

As we will present in Chapter 4, in computer graphics, the statistics of ray samples permits to identify much more rigorously than in classic image processing the pixels sharing the same model. Indeed, all ray samples hitting a given pixel and its neighbors can be used for that purpose.

Similar pixels can be detected by comparing their empirical ray color distributions using an adequate histogram distance. Since the order in which the samples are calculated is irrelevant, the sample color empirical distribution appears as a natural and complete descriptor of the compared sets. Figure 1.9 shows a small region of a Monte Carlo rendered image where two pixels are singled out. Although both pixels have different colors, their color distributions are strikingly similar and can be fused. The difference in the pixel colors may be the consequence of the presence of a single very bright ray sample in one of the distributions. By comparing the ray color distributions, it is nevertheless possible to conclude that both pixels are from the same "nature", while this conclusion could not be reached by comparing the pixel values.

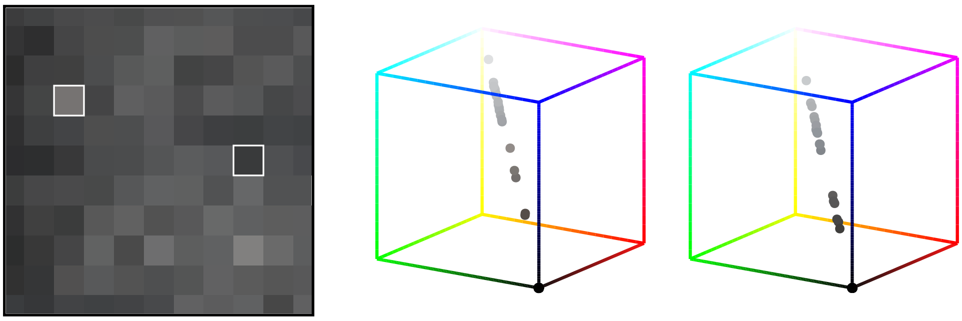


Figure 1.9: Monte Carlo rendered pixels can be grouped very efficiently by comparing their ray color distribution. Left: a crop of a Monte Carlo rendered image where two pixels with different colors are singled out. The difference in color is due to a poor estimate from a low number of rays cast at each pixel. Right: the color sample distributions of each pixel. The color sample distribution is represented in the RGB color box, where the color of each of the rays cast at a pixel is one point. The color distributions are strikingly similar and can be fused, which is the principle of the proposed algorithm.

We propose a simple but powerful filtering algorithm that uses exclusively the colors and positions of the cast ray samples. It can be thus coupled with any Monte Carlo sampler keeping a record of rays. The algorithm does not assume any noise model. It generalizes the NL-means denoiser, and shares with it an artifact-free record. The cornerstone of the proposed algorithm is to find and average the most similar patches by comparing the ray color sample distributions of each of its pixels. Inspired by this concept we name the algorithm *ray histogram fusion*. Figure 1.10 shows a running example.

In a pure Monte Carlo rendering the estimation error presents white noise characteristics.

This means that all frequencies are equally contaminated by noise. To this purpose, we introduce a multi-scale implementation that sequentially decomposes the input noisy image at each scale, filters each scale and reconstructs the multi-scale filtered image.

As shown in the experimental section of Chapter 4, the proposed filter is consistent. As the number of samples increases, more evidence is required to average two pixels. In the limit two pixels will be averaged only if their color distributions are the same. Therefore, in practice, as the number of samples grows the method converges to the expected solution. The acceleration factor depends on the degree of self similarity of the scene, which fortunately is usually high [Lebrun et al. 2012]. The algorithm provides a PSNR gain of 10 to 15 decibels, or equivalently accelerates the rendering process by using 10 to 30 times fewer samples without observable bias. It is immediately extendable to synthetic movies. Being based on the ray color values only, it can be combined with all rendering effects.

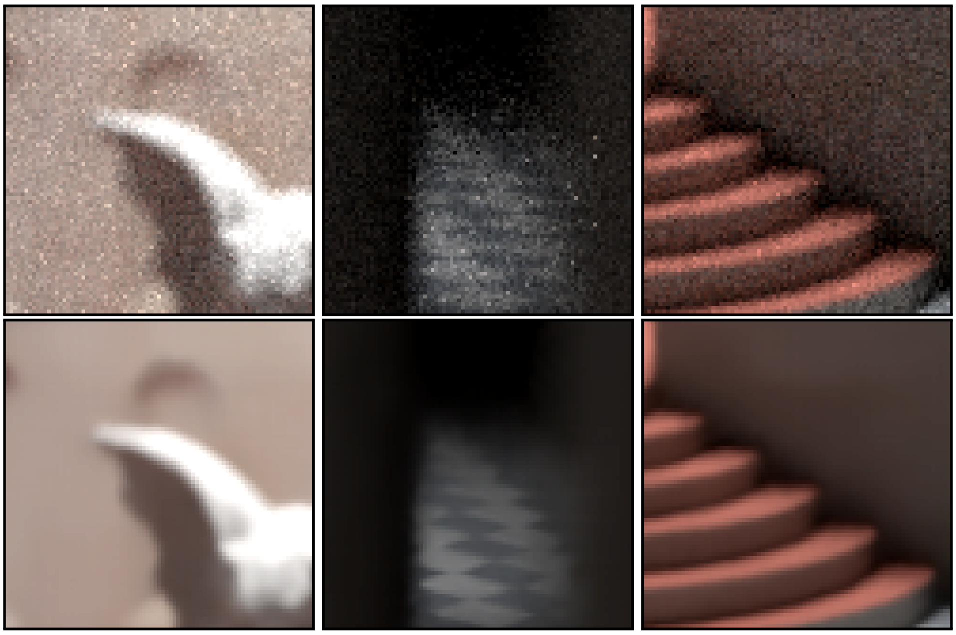


Figure 1.10: A running example of the *ray histogram fusion* algorithm. The filter increased the PSNR of this Monte Carlo rendered image by $+11.6$ decibels. To get an equal PSNR with pure Monte Carlo, $15\times$ more samples would have been needed.

Reproducible Research

Being able to reproduce experiments is a major problem in computer science. Quite often, the level of detail given in an article does not allow its complete unambiguous implementation.

In this thesis, we did our best to give a fair enough level of reproducibility. The two algorithms for the PSF estimation were published in the IPOL journal where they can be tested online. This open access journal seeks to mitigate the reproducibility problem by publishing for each article a precise algorithmic description, a reference source code and a demo facility where users can try the algorithm online.

Concerning the ray fusion histogram filter, we did our best to give enough detail so it can be completely reproduced. Unfortunately, due to the complexity of the input data, it is impossible (at least in the current scenario) to submit it to IPOL. Notwithstanding, we plan to release a reference source code with examples. This is on going work.

Summary of Contributions

Pattern based PSF estimation We present a theoretical analysis proving that the subpixel PSF estimation problem is well-posed for a single well chosen observation. Theoretical bounds show that near-optimal accuracy can be achieved with a calibration pattern formed from a realization of a random field where each pixel is an independent Bernoulli variable. We propose an algorithm that accurately registers the pattern image, normalizes its non-uniform illumination and computes the local subpixel PSF by inverting a well conditioned linear system. To the best of our knowledge, this is the first regularization-free and non-parametric local subpixel PSF estimation method reported in the literature or used in the industry.

Two-photographs PSF estimation We propose an algorithm for the subpixel estimation of the point spread function of a digital camera from aliased photographs. The proposed algorithm simply uses two fronto-parallel photographs of any planar textured scene at different distances. We develop a mathematical theory proving that the camera PSF can be derived from these two images, under reasonable conditions. Mathematical proofs supported by experimental results show the well-posedness of the problem and the convergence to the camera in-focus PSF. Experimental comparison with real camera data shows that the resulting PSF estimates reaches the accuracy levels of the best calibration pattern based state-of-the-art methods.

Boosting Monte Carlo renderers We propose a new multi-scale filter to be used as an acceleration of Monte Carlo renderers. Each image pixel is represented by the colors of the rays that reach its surface. A robust histogram distance compares the empirical color distribution associated with each pixel at each scale and decides whether two pixels can share their rays. This simple and easily reproducible algorithm provides a significant gain in PSNR, or equivalently accelerates the rendering process by using fewer samples without introducing observable bias. The algorithm is universal in the sense that it can be combined with any rendering effect. It is consistent and does not assume a particular noise model and is immediately extendable to synthetic movies.

List of Publications

The work in this thesis has led to the publication of the following articles:

- 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.
- 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, vol. 5, no. 4, pp. 1234-1260, November 2012.

- M. Delbracio, P. Musé, A. Buades, J. Chauvier, N. Phelps and J.M. Morel. *Boosting Monte Carlo rendering by distribution-driven filtering.* Submitted to Transactions on Graphics, November 2012. Under review. Conditionally accepted.
- M. Delbracio, P. Musé and A. Almansa. *Non-parametric subpixel local point spread function estimation.* Image Processing Online, 2012.
- M. Delbracio, A. Almansa and P. Musé. *Recovering the Subpixel PSF from Two Photographs at Different Distances*. Submitted to Image Processing Online, 2012.

and to the following patent application:

- J. Chauvier, M. Delbracio and N. Phelps. *A method for accelerating Monte Carlo renders.* U.S. Patent Application, Unpublished (filing date Nov. 5, 2012), no. 13/668807.

Part I

Recovering the Camera Point Spread Function

2 PSF Estimation from a Calibration Pattern Image

Most medium to high quality digital cameras (DSLRs) acquire images at a spatial rate which is several times below the ideal Nyquist rate. For this reason only aliased versions of the cameral point-spread function (PSF) can be directly observed. Yet, it can be recovered, at a subpixel resolution, by a numerical method. Since the acquisition system is only locally stationary, this PSF estimation must be local. This chapter presents a theoretical study proving that the subpixel PSF estimation problem is well-posed even with a single well chosen observation. Indeed, theoretical bounds show that a near-optimal accuracy can be achieved with a calibration pattern mimicking a Bernoulli(0.5) random noise. The physical realization of this PSF estimation method is demonstrated in many comparative experiments. We use an algorithm to accurately estimate the pattern position and its illumination conditions. Once this accurate registration is obtained, the local PSF can be directly computed by inverting a well conditioned linear system. 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 subpixel PSF estimation scheme is the first of its kind.

2.1 Introduction

Image blur can be observed when the camera focus was wrong, when there are different objects at different depths, or when there is a motion blur. But 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.

There are several key applications of PSF estimation, among them image superresolution, image de-blurring and camera quality evaluation. Traditionally sharp PSFs are considered to lead to better images, but too sharp PSFs (containing significant frequency components beyond the Nyquist frequency) cause aliasing effects that may also affect the quality of digital images. An accurate subpixel estimation of the PSF is therefore crucial to evaluate the image quality in terms of a trade-off between sharpness and aliasing effects.

Image superresolution is the longstanding problem of increasing the resolution of an aliased imaging system by interpolating a single-frame, or by fusing several low-resolution images. For this difficult superresolution process, an accurate PSF is fundamental. Surprisingly, there are many more works on blind de-convolution associated to image restoration or on superresolution, than on the accurate PSF estimation.

Existing PSF estimation methods can be classified as blind or non-blind, parametric or non-parametric. Blind methods estimate the PSF from a single image or from a set of acquired images, without any knowledge of the scene. On the contrary, non-blind methods use a specially designed calibration pattern. Blind methods endeavor to model features of the latent sharp image and to find by optimization the most suitable kernel that predicts them from the blurry observation. Most of them attempt to detect edges in the blurred image, modeling them as the result of blurring pure step-edge functions [Chalmond 1991; Luxen and Förstner 2002; Capel 2004; Smith 2006]. However, in real images, a step-edge convolved with the PSF kernel is generally not a good model of the observed edges as noted by Ladjal [2005, chapter 4]. Other blind approaches try to estimate the PSF based on statistical models of sharp images [Chalmond 1991; Rooms et al. 2004; Zhang and Cham 2008; Šroubek et al. 2007]. Since the blind estimation is an ill-posed problem (blind source separation), strong kernel smoothness assumptions or, equivalently, very simple parametric models are necessary. These inaccurate approaches are necessary to characterize and to blindly restore images affected by contingent motion or out of focus blur.

Non-blind methods instead address the problem of estimating accurately the inherent camera blur. They rely on photographs of calibration patterns to estimate the PSF. These patterns range from pin-hole or slanted-edge patterns to random noise images. The subpixel PSF estimation problem is generally treated as ill-posed. Most non-blind methods therefore introduce a PSF model constraining the space of possible solutions. Parametric models, priors on the regularity of the PSF or on its symmetry are the most current assumptions. However, these *a priori* assumptions can jeopardize the estimation accuracy.

The ideal calibration pattern that comes to mind would be a perfect pin-hole image simulating a Dirac delta impulse, permitting to directly observe samples of the PSF. However, in such an observation the signal to noise ratio would be very low, the spot support being ideally infinitesimal. Furthermore, for producing subpixel PSF estimates several subpixel-shifted versions of the spot image would be needed. Bar or sine patterns can also help sample the MTF, but only up to the Nyquist frequency.

The ISO 12233 standard [ISO 2000] gives a normalized pattern and a procedure for measuring the one-dimensional MTF, i.e., the modulus of the Fourier transform of the system's impulse response (PSF) in a particular orientation. This standard is based on the slanted-edge method [Reichenbach et al. 1991], which is an extension of the step-edge technique to achieve sub pixel resolution on the estimation. By aligning the step-edge slightly off the orthogonal scan direction the effective sampling rate is increased. Also, scan-line averaging successfully suppresses noise and increases signal-to-noise ratio making the estimation more stable. Zandhuis et al. [1997] propose a slanted-edge non-parametric subpixel PSF estimation method that admits geometrical distortions. A parametric and non-parametric edge spread function estimation procedure is proposed by Claxton and Staunton [2008]. Non-uniform illumination is also taken into account. However, the differentiation step that gives back the PSF requires regularization and therefore loses accuracy. Since the previous methods are based on estimating several one-dimensional responses, several images or symmetry assumptions are needed to reconstruct a full bi-dimensional PSF.

Before this thesis, the recent method by Joshi et al. [2008] arguably represented the state-of-the-art of slanted-edge methods. It proposes a flexible blind and non-blind non-parametric local PSF estimation algorithm. Its approach is based on the ability to detect edges with subpixel accuracy. In order to get a precise local PSF a specially designed pattern formed by 120-degrees-arc-step-edges is used. The method directly solves the de-convolution and superresolution problem for a bi-dimensional subpixel PSF. To reach a subpixel accuracy a penalty term on the norm of the PSF gradient is introduced, the inverse problem being ill-posed. As we shall see in Section 2.5, this penalty causes inaccurate estimates in the high frequency components of the PSF. If the observed image is under-sampled, which is highly probable and the reason why a subpixel PSF estimation will be proposed here, interpolating it tramples high frequency information.

As we shall try to prove, there are two main possible improvements to the Joshi et al. method, and they are linked: one is the use of a random noise pattern and the other is the removal of any regularity term, thus transforming the PSF estimation problem into a well-posed problem. The use of random noise patterns with known power spectral density has been explored for MTF estimation by Daniels et al. [1995]; Levy et al. [1999]; Backman et al. [2003, 2004]. In an ideal situation, the power spectral density $\mathrm{PSD}(f)$ of the observed digital image at frequency f is equal to the input power spectral density $\mathrm{PSD}_i(f)$ times the squared $\mathrm{MTF}(f)$. The advantage of this procedure is that the MTF can be directly calculated. It does not require knowledge of the particular noise realization, relying only on statistical assumptions. A strong limitation of this approach is that the estimation is done up to half the sampling frequency. Consequently it does not reach a subpixel accuracy, and aliasing effects are not taken into account.

Brauers et al. [2010] also used a random noise pattern, but in a completely different approach. The acquired image is registered to match the pattern. Then, by doing de-convolution with the almost flat spectrum noise pattern, this method succeeds in characterizing locally the PSF. However, the method assumes that the camera over-samples the signal, which is a correct hypothesis for the particular multi-spectral-camera-lens system, but unrealistic for a classical optical camera. This method contemplates the possibility of a non-linear light sensor response, but does not correct the non-projective distortion. Again, the question is treated as an ill-posed problem and noise-free kernels are produced by regularization.

Table 2.1 summarizes some of the existing algorithms for PSF estimation. It first gives the abbreviations for the five criteria characterizing calibration methods. The above analysis suggests that an ideal method must be non-blind (NB), with no regularization. The kernel estimation must be 2D, local (L), subpixel (sp). The main systematic perturbations in imaging (optical distortion (D), non uniform illumination (I), non linear sensor response (G)) must be corrected when comparing the ideal pattern to the photographed one. In short, an ideal method must be (NB,R,2D-L-sp, DIG) with no (C,P,K). The closest to this ideal in the state-of-the-art was the Joshi et al. method, but it includes a regularization which will be shown fatal to the high frequency kernel content. The method proposed here has all "good" features. Indeed, it will be shown mathematically and practically that an adequate noise pattern permits to avoid any regularization. The camera kernel is directly recovered from the comparison of the ideal noise pattern to the observed one by the inversion of a well-conditioned matrix. We will also verify that this is not possible with an edge based pattern.

By correctly choosing the calibration pattern, a subpixel PSF estimation is therefore feasible without *a priori* kernel model, without regularization, and with a single aliased input image capture. Nevertheless, this requires the careful correction of the geometrical distortion, of the non-uniform illumination, and of the non-linearity of the sensor response. In short, with a noise pattern, and thanks to this careful elimination of all bias, the PSF subpixel estimation becomes well-posed. Theoretical bounds will also demonstrate the quasi-optimality of white noise calibration patterns to that purpose. Given that the PSF is space variant, due to lens aberrations or sensor non uniformity, the estimation must be done as local as possible.

This chapter is organized as follows. Section 2.2 describes the general mathematical digital camera model used for PSF estimation method. Section 2.3 proposes a mathematical theory of

Item	Code	Description
Blindness	B	blind
	NB	non-blind
Model	E	edge-based
	R	random pattern
	N	natural image model
Regularity	C	circular symmetry
	P	parametric estimation
	K	other kernel regularization
Estimation	1	one-image estimation
	k	k-image estimation
	1D, 2D	uni/bi-dimensional estimation
	L	local estimation
	sp	subpixel estimation
Features	D	geometrical distortion considered
	I	non-uniform illumination considered
	G	non-linear sensor response considered

Algorithm	Blind	Model	Regul.	Estim.	Feat.
Luxen and Förstner [2002]	B	E	P	1-2D	-
Smith [2006] Capel [2004]	B	E	P	1-1D	-
Rooms et al. [2004] Zhang and Cham [2008] Šroubek et al. [2007]	B	N	C-P	1-2D	-
Chalmond [1991]	B	E-N	K	1-2D	-
Zandhuis et al. [1997]	NB	E	-	k-2D-L-sp	D
Claxton and Staunton [2008]	NB	E	P-K	k-2D-sp	I
Reichenbach et al. [1991]	NB	E	-	1-1D-sp	-
Joshi et al. [2008]	NB	E	K	1-2D-L-sp	D
Daniels et al. [1995] Levy et al. [1999] Backman et al. [2004, 2003]	NB	R	C	1-1D	-
Brauers et al. [2010]	NB	R	K	1-2D-L	G
Proposed	NB	R	-	1-2D-L-sp	DIG

Table 2.1: PSF estimation algorithm summary.

optimal patterns. It studies the optimality of the calibration pattern in terms of the well-posedness of the PSF estimation problem, and concludes with the proposition of a near optimal and physically feasible random noise pattern. Section 2.4 describes all the steps of the proposed PSF estimation protocol. In Section 2.5 experimental results generated with both simulated and real camera data are presented, cross-validated, and compared with the results of state of the art previous methods. Section 2.6 is a final discussion.

2.2 Image Formation 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 three-dimensional 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 anti-aliasing filters, 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

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

where $F(\cdot)$ is the geometric distortion operator, h is the convolution kernel due to all intrinsic PSF-like effects, h_{ex} is the convolution due to extrinsic blurring effects that occur outside the camera (like motion blur and atmospheric turbulence), 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 $\mathbf{n}(u)$ models the sensor noise.

Physical models of digital camera sensors, both for CCD and CMOS sensors, suggest that the readout noise $\mathbf{n}(u)$ is a mixture of luminance independent (Gaussian, thermal) noise, and luminance dependent (Poisson or photon counting) noise [Healey and Kondepudy 1994; Tian et al. 2001; Marion 1997]. In fact, the noise can be modeled as white Gaussian noise with luminance-dependent variance. For the purposes of this study, however, precise statistics of noise are not critical, and only the global SNR is significant, so we shall stick to the more traditional and simpler model of image independent white Gaussian noise.

Furthermore, as stated in the introduction, we will only deal here with intrinsic PSF-like effects at the camera focal plane. Therefore we assume and that the experimental setup is capable of avoiding motion blur, atmospheric turbulence and out-of-focus blur as much as possible, thus permitting to neglect the effect of h_{ex}. Strictly speaking out-of-focus blur should be included in the intrinsic camera blur kernel h, but since it can become negligible under controlled experimental setups, we chose to exclude this kind of PSF-like effect from our study.

As a result of the previous discussion we shall simplify model (M$'$) and use

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

where h models the camera-intrinsic PSF-like blurring effects, and $\mathbf{n}$ is an image-independent Gaussian white noise.

The blur kernel h is space variant, but it varies 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 implicit in the combined deformation operator $F \circ H$ 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 becomes

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

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 shall first focus on the simplified formation model

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

and explain later on how g can be eliminated. The next section discusses the structure of the optical kernel h.

2.2.1 Diffraction-Limited Optical Systems

Ideal optical systems present PSFs only caused by the optical light diffraction. In the case where there are no aberrations the diffraction kernel is determined by the shape and size of the aperture, the focal length, and the wavelength of the considered monochromatic light. If the shape and size of the aperture is known, the far field approximation (*Fraunhofer diffraction*) can be explicitly computed as the square of the Fourier transform modulus of the aperture function [Goodman 1996]. *As a trivial consequence the* PSF *diffraction kernel is always non-negative.*

In the reasonable, though inexact hypothesis that the aperture is circular, the diffraction kernel writes

$$h_{\text{diff}}(\mathbf{x}) = \left(\frac{A}{\lambda f}\right)^2 \cdot \left[\frac{2J_1(r)}{r}\right]^2,$$

with $r = \frac{\pi D|\mathbf{x}|}{\lambda f}$ and $A = \pi(D/2)^2$. The function $J_1(r) = \frac{1}{\pi}\int_0^\pi \cos(\theta - r\sin\theta)d\theta$ is the Bessel function of the first kind and order one, f is the lens focal length, D the aperture diameter, and λ the wavelength. In the case of a circular aperture, the cutoff frequency of a diffraction limited system is $\rho_c = \frac{2\pi D}{\lambda f}$. This frequency depends only on the so called F-number $= \frac{f}{D}$ and the wavelength λ.

Optical aberrations degrade this ideal system where only diffraction is considered, producing larger kernels [Williams and Becklund 2002]. In addition, optical anti-aliasing filters - OLPF may be introduced in the camera before sampling. They are typically made of several birefringent crystals that separate a light spot into several divergent light spots, leading to an effect similar to having a larger pixel pitch. An analysis of the filters commonly used in digital cameras can be found in [Zhao et al. 2006].

In most cameras, the digitization process is performed by a rectangular grid of photo-sensors (CCD or CMOS) located on the focal plane. Each photo-sensor integrates the light arriving at a particular exposure time. This sensor light integration can be modeled by a convolution with a kernel $h_{\text{sensor}} = \mathbf{1}_C$, the indicator function of the photo-sensor region C. Yadid-Pecht [2000] performs a theoretical analysis of the MTF for the active area shape and deduces explicit formulas for the transfer function for CMOS pixel arrays with square, rectangular and L shaped active areas, which are regularly used. In conclusion, the unknown kernel h results from the convolution of some three different kernels, all nonnegative. Most digital cameras have only one sensor array. In order to acquire color information, each photo-sensor is filtered to capture only wavelengths of a particular band for the red, green, or blue channels which is done by a color filter array (CFA). The most popular CFA is the Bayer filter mosaic, which covers the sensor plane with 50% of green filters and 25% of blue and red filters respectively (see Figure 2.1). The image formed by the data as it comes directly off the sensor array is called RAW image.

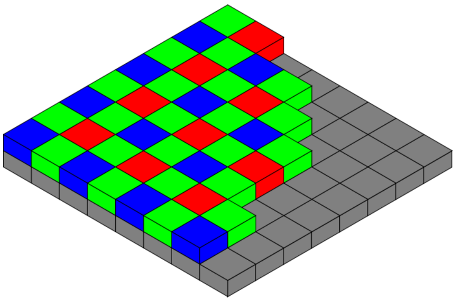


Figure 2.1: Typical Bayer pattern. Image taken from Wikipedia [2012].

In a typical configuration of $f/D = 5.6$ the diffraction cutoff frequency for the green light ($\lambda = 530$ nm) is $\rho_c = \frac{2\pi}{530\times10^{-9}\times5.6}$. Hence, to avoid aliasing the inter pixel distance should be at least $\delta_c = \frac{\pi}{\rho_c} = \frac{530\times10^{-9}\times5.6}{2}$. Assume we have a digital camera with a Bayer sensor of size 22.2 mm $\times$ 14.8 mm with a resolution of 3888×2592 pixels. This leads to an inter pixel distance $\delta_s = \frac{22.2\times10^{-3}}{3888/2}$. Thus, in order to avoid aliasing in this diffraction limited system we would need to get samples at $s = \delta_s/\delta_c \approx 7.7\times$ the camera resolution. In practice the PSF cutoff frequency will be much smaller due to the anti-aliasing filter and the light integration in the sensor.

2.3 Optimality Criterion and Quality Measure for Calibration Patterns

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. In this section we show that a nearly optimal conditioning is obtained when the calibration pattern is a realization of a white noise. While this may not be new (noise patterns have been used in non-blind PSF estimation, see e.g. [Daniels et al. 1995; Levy et al. 1999; Backman et al. 2004, 2003; Brauers et al. 2010]), the novelty presented in this section 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 as a white noise image is chosen as the calibration pattern.*

2.3.1 Inverse problem statement in terms of the re-sampling operator and the calibration pattern

In the following, $\mathbf{F}$ denotes the Fourier transform and $\hat{f}$ the Fourier transform of a function f. The s-Shannon-Whittaker interpolator defined as $I_s\mathbf{u}(\mathbf{x}) = \sum_{\mathbf{k}} \mathbf{u}(\mathbf{k}) \operatorname{sinc}(s^{-1}\mathbf{x} - \mathbf{k})$ is denoted by I_s, $\mathbf{S}_s$ is the s-over-sampling operator $\mathbf{S}_s u(\mathbf{k}) = u(s^{-1}\mathbf{k})$ and LPF_w is the frequency cut-off low pass filter that cuts the spectrum of a signal to $[-w\pi, w\pi]^2$.

Suppose that h is band-limited within $\operatorname{supp}(\hat{h}) = [-\delta\pi, \delta\pi]^2$. If the PSF is sampled at a rate s, where $s > \delta$, the Nyquist sampling theorem guarantees a perfect signal reconstruction. We will consider the case where $\delta > 1$, which corresponds to aliased images, as in practice most digital cameras introduce aliasing.

Lemma 1 (Discrete Convolution)**.** *Let u and h be images in $L^2(\mathbb{R}^2)$ such that h is band-limited, i.e., supp$(\hat{h}) = [-s\pi, s\pi]^2$. Then*

$$u * h = I_s\left(\tilde{\mathbf{u}} * \mathbf{h}\right),$$

where $\mathbf{h} = \mathbf{S}_s h$ and

$$\tilde{\mathbf{u}} = \mathbf{S}_s \text{LPF}_s u.$$

Proof. Set $\tilde{u} := \text{LPF}_s u = \mathbf{F}^{-1}\left(\hat{u} \cdot \mathbf{1}_{[-s\pi, s\pi]^2}\right)$, so that $\tilde{\mathbf{u}} = \mathbf{S}_s \tilde{u}$ and $\hat{\tilde{u}} = \hat{u} \cdot \mathbf{1}_{[-s\pi, s\pi]^2}$. This implies that $u * h = \tilde{u} * h$. Indeed,

$$\mathbf{F}^{-1}\left(\hat{u}\hat{h}\right) = \mathbf{F}^{-1}\left(\hat{u} \cdot \hat{h} \cdot \mathbf{1}_{[-s\pi, s\pi]^2}\right) = \mathbf{F}^{-1}\left(\hat{\tilde{u}}\hat{h}\right).$$

Now, since both $\hat{\tilde{u}}$ and $\hat{h}$ are supported in $[-s\pi, s\pi]^2$, it follows that

$$\begin{aligned} \tilde{u} * h &= I_s\mathbf{S}_s(\tilde{u}) * I_s\mathbf{S}_s(h) \\ &= I_s(\tilde{\mathbf{u}}) * I_s(\mathbf{h}) \\ &= I_s(\tilde{\mathbf{u}} * \mathbf{h}). \end{aligned}$$

$\square$

Remark 1. *Note that u does not need to be band-limited, only h. Notwithstanding, if we can find the spectral cut-off $\tilde{u}$ of u, then this lemma implies that the continuous convolution $u * h$ can be simulated exactly with a* discrete *set of samples.*

Let us denote by $\mathcal{S}_s$ the s-to-1-sub-sampling operator

$$\mathcal{S}_s = \mathbf{S}_1 I_s.$$

It follows from Lemma 1 that the image formation model (M) can be rewritten in terms of discrete sequences as

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

where $\mathbf{h}$ and $\tilde{\mathbf{u}}_D$ are sampled at rate s such that $s > \delta$ for $\mathbf{h}$ to be well sampled. The value s is the over-sampling rate to the high resolution lattice, where the PSF estimation is going to take place, from the $1\times$ sensor grid.

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

$$\arg\min_{\mathbf{h}} \quad \|\mathcal{S}_s \tilde{\mathbf{u}}_D * \mathbf{h} - \mathbf{v}\|_2^2 \tag{2.1}$$

Here, $\tilde{\mathbf{u}}_D$ is the result of the Shannon-sampling on the $s\times$ grid of the distorted continuous pattern signal $\tilde{\mathbf{u}}_D = \mathbf{S}_s \text{LPF}_s u(D(\mathbf{x}))$, and $\mathbf{v}$ the blurred degraded digital observation on the camera $1\times$ sensor grid.

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

- to choose a good PSF characterization pattern;
- to estimate the function $g(\cdot)$, the non-linear sensor response;
- to estimate the geometric deformation $D(\cdot)$;
- to generate $\tilde{\mathbf{u}}_D$ from the sharp latent pattern image u;
- to find numerical algorithms calculating 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. Strictly speaking h being band-limited cannot be compactly supported. However, the error introduced by a restriction on the support will prove negligible in comparison to the other sources of error: image noise, quantization, slight estimation errors of g, D,... The found solution $\mathbf{h}$ is experimentally independent from variations of its assumed support.

The problem in (2.1) can be rewritten in matrix form,

$$\arg\min_{\mathbf{h}} \quad \|\mathbf{S}_s \mathbf{C}[\tilde{\mathbf{u}}_D]\mathbf{h} - \mathbf{v}\|_2^2, \tag{P}$$

where $\mathbf{C}[\tilde{\mathbf{u}}_D]$ 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 sizes of $\tilde{\mathbf{u}}_D$ and $\mathbf{h}$ are $ms \times ns$ and $r \times r$, respectively. 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 δ, which facilitates the construction of the subsampling matrix $(\mathbf{S}_s\mathbf{u})(m,n) = \mathbf{u}(ms, ns)$. Then $\mathbf{S}_s\mathbf{C}[\tilde{\mathbf{u}}_D]$ is of size $M \times N$, with $N = r \times r$.

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

$$\mathbf{h}_e = (\mathbf{S}_s\mathbf{C}[\tilde{\mathbf{u}}_D])^+\mathbf{v},$$

where

$$(\mathbf{S}_s\mathbf{C}[\tilde{\mathbf{u}}_D])^+ = \left((\mathbf{S}_s\mathbf{C}[\tilde{\mathbf{u}}_D])^t(\mathbf{S}_s\mathbf{C}[\tilde{\mathbf{u}}_D])\right)^{-1}(\mathbf{S}_s\mathbf{C}[\tilde{\mathbf{u}}_D])^t$$

is the Moore-Penrose pseudo-inverse of $(\mathbf{S}_s\mathbf{C}[\tilde{\mathbf{u}}_D])$. Depending on the condition number of this matrix, the inversion would be well-posed and the solution would be unique. Since

$$(\mathbf{S}_s\mathbf{C}[\tilde{\mathbf{u}}_D])\mathbf{h} + \mathbf{n} = \mathbf{v},$$

the estimation error is given by $\mathbf{n}_e = (\mathbf{S}_s\mathbf{C}[\tilde{\mathbf{u}}_D])^+\mathbf{n}$. The noise has zero-mean, thus the estimator $\mathbf{h}_e$ is unbiased and its variance is

$$\begin{aligned} E\left\{\|\mathbf{n}_e\|_2^2\right\} &= E\left\{\left\|(\mathbf{S}_s\mathbf{C}[\tilde{\mathbf{u}}_D])^+\mathbf{n}\right\|_2^2\right\} \\ &= E\left\{\sum_{j=1}^{M}\left(\sum_{i=1}^{N}(\mathbf{S}_s\mathbf{C}[\tilde{\mathbf{u}}_D])^+_{ij}n_i\right)^2\right\} \\ &= \sum_{j=1}^{M}\sum_{i=1}^{N}\sum_{k=1}^{N}(\mathbf{S}_s\mathbf{C}[\tilde{\mathbf{u}}_D])^+_{ij}(\mathbf{S}_s\mathbf{C}[\tilde{\mathbf{u}}_D])^+_{kj}E\left\{n_in_k\right\}. \end{aligned}$$

Since $\mathbf{n}$ is white and stationary, with zero mean, it follows that

$$E\left\{\|\mathbf{n}_e\|_2^2\right\} = \sum_{j=1}^{M}\sum_{i=1}^{N}(\mathbf{S}_s\mathbf{C}[\tilde{\mathbf{u}}_D])^{+2}_{ij}\sigma_n^2 = \|(\mathbf{S}_s\mathbf{C}[\tilde{\mathbf{u}}_D])^+\|_F^2\sigma_{\mathbf{n}}^2,$$

where $\sigma_{\mathbf{n}}^2$ denotes the noise variance, and $\|\cdot\|_F$ is the Frobenius norm of a matrix.

If all singular values of $\mathbf{S}_s\mathbf{C}[\tilde{\mathbf{u}}_D]$ are non zero, the singular values of $(\mathbf{S}_s\mathbf{C}[\tilde{\mathbf{u}}_D])^+$ are the inverses of the singular values of $(\mathbf{S}_s\mathbf{C}[\tilde{\mathbf{u}}_D])$. If some singular value is zero, the system is ill posed and the estimation problem cannot be solved, unless some kind of regularization on $\mathbf{h}$ is imposed.

Let $\{\sigma_1, \sigma_2, \ldots, \sigma_N\}$ be the singular values of $\mathbf{S}_s\mathbf{C}[\tilde{\mathbf{u}}_D]$. Then

$$\|(\mathbf{S}_s\mathbf{C}[\tilde{\mathbf{u}}_D])^+\|_F^2 = \sum_{i=1}^{N}\sigma_i^{-2}.$$

In order to minimize the variance of the estimator $\mathbf{h}_e$ (i.e., to minimize the noise amplification), one has to minimize the function

$$\gamma(\mathbf{S}_s\mathbf{C}[\tilde{\mathbf{u}}_D]) := \sum_{i=1}^{N}\sigma_i^{-2}$$

It should be pointed out that γ depends on the rate s and on the samples $\tilde{\mathbf{u}}_D$. The superresolution rate s is determined by the spectral support of the PSF. The sequence $\tilde{\mathbf{u}}_D$ depends on the adopted continuous pattern u, on the geometric transformation D (that includes the perspective projection associated to the particular pattern's view) and also to other possible distortions presented in the camera-lens system. Hence, for the $s\times$ subpixel PSF estimation problem, γ measures the quality of any given **view** of a calibration pattern.

In order to find the best ideal pattern independently of the view and distortion, we will consider first the discrete problem of finding the best sequence $\tilde{\mathbf{u}}_D$, minimizing the γ value. To simplify the notation we write $u_{ij} = (\tilde{\mathbf{u}}_D)_{ij}$. This motivates the following definition.

Definition 1 (Optimal digital pattern). *Given a kernel support $N = r \times r$ and a window observation size $M = m \times n$, the optimal pattern for the $s\times$ subpixel PSF estimation is the digital calibration pattern $\mathbf{u}^*$ such that*

$$\mathbf{u}^* = \underset{a \le u_{ij} \le b}{\arg\min}\, \gamma(\mathbf{S}_s\mathbf{C}[\mathbf{u}]).$$

where the constraints on u_{ij} are linked to the physical realization of the pattern and to the sensibility of the sensors. (The conclusions of the analysis will prove independent of the particular value of these bounds.)

2.3.2 Characterization of optimal digital calibration patterns

In this section, we derive a lower bound for $\gamma(\mathbf{S}_s\mathbf{C}[\mathbf{u}])$ that will be used to design calibration patterns. Indeed, it will then be shown that for a realization of white stationary Bernoulli noise, the γ value is so close to this bound, that in practice these patterns can be considered to be optimal.

Lemma 2. *Let $\mathbf{\Phi}$ be a $M \times N$ matrix, $M > N$, with all its entries in $[a, b]$. Let $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_N$ denote its singular values. Then*

$$\sum_{i=2}^{N} \sigma_i^2 \leq (a+b)\sqrt{MN}\sigma_1 - \sigma_1^2 - abMN.$$

Proof. Let φ_{ij} be the (i, j) entry of $\mathbf{\Phi}$, and $\boldsymbol{\varphi}_j$ its j-th column. Let also $\bar{\boldsymbol{\varphi}}_j = \frac{1}{M}\sum_{i=1}^{M}\varphi_{ij}$ denote each column's mean, and $\hat{\boldsymbol{\varphi}}_j = \boldsymbol{\varphi}_j - \bar{\boldsymbol{\varphi}}_j\mathbf{1}$.

The Frobenius norm of $\mathbf{\Phi}$ can be expressed as

$$\|\mathbf{\Phi}\|_F^2 = \text{trace}(\mathbf{\Phi}^t\mathbf{\Phi}) = \sum_{j=1}^{N} \boldsymbol{\varphi}_j^t \boldsymbol{\varphi}_j.$$

Since $\tilde{\varphi}_{ij} := \frac{\varphi_{ij}-a}{b-a} \in [0, 1]$, we have $\tilde{\varphi}_{ij}^2 \leq \tilde{\varphi}_{ij}$, and then

$$\boldsymbol{\varphi}_j^t \boldsymbol{\varphi}_j = \sum_{i=1}^{M} \varphi_{ij}^2 \leq \sum_{i=1}^{M} (a+b)\varphi_{ij} - abM = M(a+b)\bar{\varphi}_j - abM$$

Thus,

$$\|\mathbf{\Phi}\|_F^2 \leq M(a+b)\sum_{j=1}^{N} \bar{\varphi}_j - abMN. \tag{2.2}$$

On the other hand, for all $\mathbf{x}$ such that $\|\mathbf{x}\| = 1$, $\|\mathbf{\Phi}\|_2 \geq \|\mathbf{\Phi}\mathbf{x}\|$. Let us take $\mathbf{x} = \frac{1}{\sqrt{N}}\mathbf{1}$. Then

$$\begin{aligned}
\|\mathbf{\Phi}\|_2^2 \geq \|\mathbf{\Phi}\mathbf{x}\|_2^2 &= \frac{1}{N}\Big\|\sum_{j=1}^{N} \boldsymbol{\varphi}_j\Big\|^2 \\
&= \frac{1}{N}\Big\|\sum_{j=1}^{N} \hat{\boldsymbol{\varphi}}_j + \sum_{j=1}^{N} \bar{\varphi}_j\mathbf{1}\Big\|^2 \\
&= \frac{1}{N}\left(\Big\|\sum_{j=1}^{N} \hat{\boldsymbol{\varphi}}_j\Big\|^2 + \Big\|\sum_{j=1}^{N} \bar{\varphi}_j\mathbf{1}\Big\|^2\right) + \\
&\quad \frac{1}{N}\left(\sum_{j=1}^{N} \hat{\boldsymbol{\varphi}}_j\right)^t \left(\sum_{j=1}^{N} \bar{\varphi}_j\mathbf{1}\right) \\
&= \frac{1}{N}\left(\Big\|\sum_{j=1}^{N} \hat{\boldsymbol{\varphi}}_j\Big\|^2 + \Big\|\sum_{j=1}^{N} \bar{\varphi}_j\mathbf{1}\Big\|^2\right) \\
&\geq \frac{1}{N}\Big\|\sum_{j=1}^{N} \bar{\varphi}_j\mathbf{1}\Big\|^2 = \frac{M}{N}\left(\sum_{j=1}^{N} \bar{\varphi}_j\right)^2.
\end{aligned} \tag{2.3}$$

Thus, by (2.2) and (2.3), we have:

$$\|\mathbf{\Phi}\|_2 \geq \frac{\|\mathbf{\Phi}\|_F^2 + abMN}{\sqrt{MN}(a+b)}.$$

Then, since $\|\mathbf{\Phi}\|_2 = \sigma_1$,

$$\sigma_1 \geq \frac{\|\mathbf{\Phi}\|_F^2 + abMN}{\sqrt{MN}(a+b)} = \frac{\sum_{i=1}^N \sigma_i^2 + abMN}{\sqrt{MN}(a+b)}$$

Finally,

$$\sum_{i=2}^N \sigma_i^2 \leq (a+b)\sqrt{MN}\sigma_1 - \sigma_1^2 - abMN.$$

□

Lemma 3 (A bound on γ). *Let $\mathbf{\Phi}$ be a $M \times N$ matrix, $M > N$, with all its entries φ_{ij} in $[a,b]$. Then*

$$\min_{\varphi_{ij}\in[a,b]} \gamma(\mathbf{\Phi}) \geq \frac{1}{MN}\left(\frac{1}{b^2} + \frac{4(N-1)^2}{(b-a)^2}\right).$$

Proof. According to Lemma 2, for any matrix $\mathbf{\Phi}$ with entries in $[a,b]$, and in particular for the ones that attain

$$\gamma^* = \min_{\sigma_1,\dots,\sigma_N} \sum_{i=1}^N \sigma_i^{-2},$$

the inequality $\sum_{i=2}^N \sigma_i^2 \leq (a+b)\sqrt{MN}\sigma_1 - \sigma_1^2 - abMN$ holds. Thus

$$\min_{\varphi_{ij}\in[a,b]} \gamma(\mathbf{\Phi}) \geq \min_{\sigma\in\mathcal{D}} f(\sigma),$$

where $\sigma = (\sigma_1, \dots, \sigma_N)$,

$$\mathcal{D} := \{\sigma \mid \sigma_i \geq 0,\ \sum_{i=1}^N \sigma_i^2 - (a+b)\sqrt{MN}\sigma_1 + abMN \leq 0\},$$

and $f(\sigma) := \sum_{i=1}^N \sigma_i^{-2}$. The function f being strictly convex on $\mathcal{D}$, which is itself a convex and compact domain, it follows that the minimum of f on $\mathcal{D}$ is attained at a unique point. $\mathcal{D}$ and f being invariant by any permutation of $\sigma_2, \dots, \sigma_N$, the minimum point being unique satisfies $\sigma_2 = \cdots = \sigma_N$. Since this minimum belongs to $\mathcal{D}$,

$$\sum_{i=2}^N \sigma_i^2 = (N-1)\sigma_2^2 \leq (a+b)\sqrt{MN}\sigma_1 - \sigma_1^2 - abMN.$$

By noting that the maximum value of

$$\sigma_1 \mapsto (a+b)\sqrt{MN}\sigma_1 - \sigma_1^2 - abMN$$

is $(\frac{b-a}{2})^2 MN$, it follows that

$$\sigma_2^2 \leq \left(\frac{b-a}{2}\right)^2 \frac{MN}{N-1}.$$

On the other hand for any point of $\mathcal{D}$ we have

$$\sigma_1^2 \leq (a+b)\sqrt{MN}\sigma_1 - abMN.$$

Then, it follows that $\sigma_1^2 \leq b^2 MN$. Consequently,

$$\min_{\sigma \in \mathcal{D}} f(\sigma) = \sum_{i=1}^{N} \sigma_i^{-2} \geq \frac{1}{MN}\left(\frac{1}{b^2} + \frac{4(N-1)^2}{(b-a)^2}\right).$$

□

Remark 2. *It should be noted that in the proof of the previous lemma, the condition that the entries of* $\mathbf{\Phi}$ *belong to* $[a,b]$ *was replaced by the weaker condition given by the inequality proved in Lemma 2. This amounts to enlarge the space of matrices that was originally considered, thus the real optimum that can be attained by matrices with entries in* $[a,b]$ *will necessarily lead to higher values of* γ.

Remark 3. *Notice also that in Lemma 3 we did not solve the complete constrained optimization problem*

$$\min_{\sigma_1,\ldots,\sigma_N} \sum_{i=1}^{N} \sigma_i^{-2} \quad \text{subject to} \quad \sum_{i=2}^{N} \sigma_i^2 \leq (a+b)\sqrt{MN}\sigma_1 - \sigma_1^2 - abMN.$$

While this problem can be solved via *the Karush-Kuhn-Tucker conditions, according to the previous remark it would still lead to a lower bound on* γ. *The solution of this constrained minimization problem leads to a closed form which is significantly less handy than the bound that was obtained in Lemma 3, and is worthless since both bounds are extremely close, as shown in Figure 2.2 for* $a = 0, b = 1$.

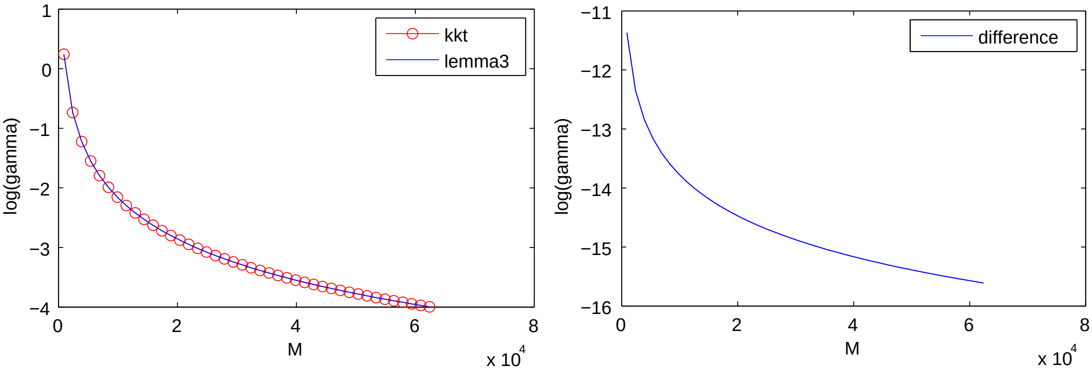


Figure 2.2: Comparison of the lower bound given by Lemma 3 and the one obtained by solving the KKT conditions for the case $a = 0, b = 1$. Both bounds are shown in the plot on the left, as a function of M. The plot at right shows their difference, also as a function of M.

Proposition 1 (Non-asymptotic bound for optimal patterns). *Let* $\mathbf{u} = \{u_{ij}\}$ *be a* $ms \times ns$ *digital image with all its values in* $[a,b]$. *Let* $\mathbf{S}_s\mathbf{C}[\mathbf{u}]$ *be the operator associated to the convolution of the* $r \times r$ *kernel with the image* $\mathbf{u}$, *followed by the downsampling operator of rate* s. *Then*

$$\min_{a \leq u_{ij} \leq b} \gamma(\mathbf{S}_s\mathbf{C}[\mathbf{u}]) \geq \frac{1}{MN}\left(\frac{1}{b^2} + \frac{4(N-1)^2}{(b-a)^2}\right)$$

where $M = m \times n$ *is the observation window size and* $N = r \times r$ *is the kernel size .*

Proof. The result follows directly from Lemma 3, since the operator $\mathbf{S}_s\mathbf{C}[\mathbf{u}]$ associated to $\mathbf{u}$ is a $M \times N$ matrix with all its entries in $[a, b]$. $\square$

We will propose as calibration pattern a realization of a white Bernoulli(0.5) stationary noise. It will be shown that this calibration pattern is so close to the $\gamma(\mathbf{S}_s\mathbf{C}[\mathbf{u}])$ lower bound given by Lemma 3 that for practical calibration purposes, it can considered to be optimal.

The motivation for choosing stationary white noise patterns is not new: white noise has been widely used for system identification applications, since it optimizes the minimum variance of unbiased estimators. Now, the choice of Bernoulli(0.5) distribution can be explained as follows. Suppose $\mathbf{u} = \{u_{ij}\}$, where $u_{ij} \in [a, b]$ are mutually independent random variables, identically distributed with mean m_u and variance σ_u^2. In this case, it can easily be shown that

$$E\left\{(\mathbf{S}_s\mathbf{C}[\mathbf{u}])^t(\mathbf{S}_s\mathbf{C}[\mathbf{u}])\right\} = M\left(m_u^2 \mathbf{1}\mathbf{1}^t + \sigma_u^2 \mathbf{I}\right).$$

This is a direct consequence of the non-correlated nature of $\mathbf{u}$ and that subsamples of white noise remain white noise. Observe that $M\left(m_u^2 \mathbf{1}\mathbf{1}^t + \sigma_u^2 \mathbf{I}\right)$ has only two different eigenvalues: $\sigma_1 = M(Nm_u^2 + \sigma_u^2)$ and $\sigma_2 = \cdots = \sigma_N = M\sigma_u^2$. Thus, its γ value is

$$\gamma = \frac{1}{M}\left(\frac{1}{Nm_u^2 + \sigma_u^2} + \frac{N-1}{\sigma_u^2}\right).$$

On the one hand, in order to minimize γ, m_u and σ_u^2 values should be as large as possible. On the other hand there is a trade-off between both values and they cannot be simultaneously maximized. Indeed, any random variable with support $[a, b]$ satisfies

$$\sigma_u^2 \leq (m_u - a)(b - m_u).$$

Nonetheless, the equality holds for the Bernoulli distribution. Hence, from now on we restrict the analysis to the Bernoulli case which, from the previous reason, is optimal. Therefore we can express γ as

$$\gamma = \frac{1}{M}\left(\frac{1}{Nm_u^2 + (m_u - a)(b - m_u)} + \frac{N-1}{(m_u - a)(b - m_u)}\right).$$

It can be shown that the $m_u \in [a, b]$ value where γ attains its minimum is always very close to $m_u = \frac{a+b}{2}$. This happens independently of M and N. However, the exact value depends on N. It is therefore convenient, to avoid dependence on N, to fix $m_u = \frac{a+b}{2}$ by using an equiprobable Bernoulli distribution. Finally, the γ value for the expected operator $\mathbf{S}_s\mathbf{C}[\mathbf{u}]$ when using a Bernoulli(0.5) pattern is

$$\gamma = \frac{4}{M}\left(\frac{1}{N(a+b)^2 + (b-a)^2} + \frac{N-1}{(b-a)^2}\right).$$

This value is very close to the bound provided by Lemma 3. Indeed, for $M \geq N \gg 1$ we have $\gamma^* - \gamma \approx \frac{4}{M(b-a)^2}$. This small difference is illustrated in Figure 2.3 for the particular case $a = 0, b = 1$. Notice also that since

$$E\left\{(\mathbf{S}_s\mathbf{C}[\mathbf{u}])^t(\mathbf{S}_s\mathbf{C}[\mathbf{u}])\right\} = \lim_{M\to\infty}(\mathbf{S}_s\mathbf{C}[\mathbf{u}])^t(\mathbf{S}_s\mathbf{C}[\mathbf{u}]),$$

large M values may be required in order to reach the optimal γ. However, this is clearly not our case of interest, our goal being to perform a local kernel estimation. Nevertheless, we may still be

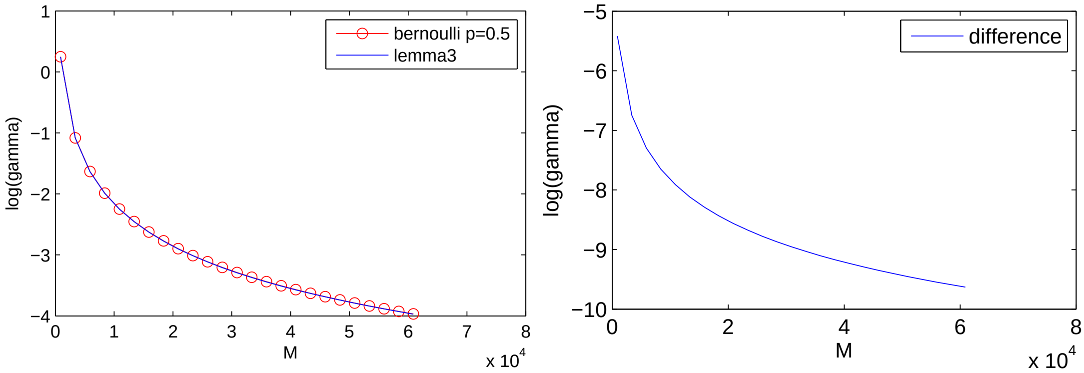
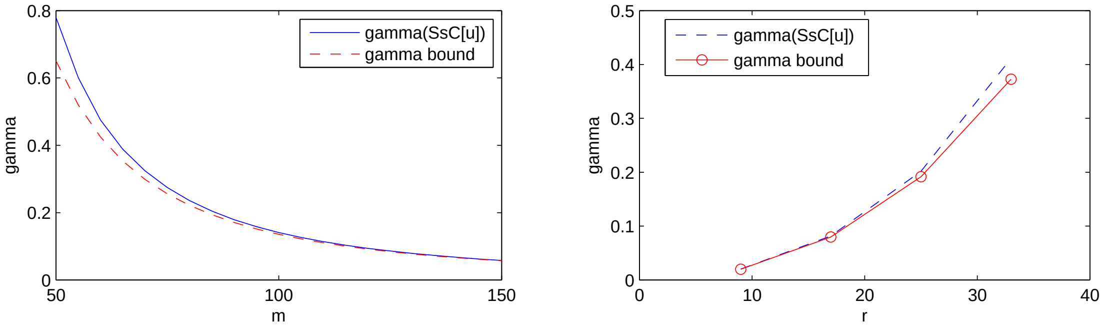


Figure 2.3: Comparison of the lower bound given by Lemma 3 and the gamma obtained from the *expected* $\mathbf{S}_s\mathbf{C}[\mathbf{u}]$ operator when using a Bernoulli(0.5) random noise pattern. Both bounds are shown in the plot on the left, as a function of M whereas the plot at right shows their difference, also as a function of M.

interested in exploring the use of a realization of white stationary Bernoulli(0.5) noise as calibration pattern, for finite and realistic values of M and N (the non-asymptotic case).

In order to show that the choice of such a calibration pattern can be considered to be optimal for practical PSF estimation, we generated a white random binary image $u_{ij} \in \{0, 1\}$, Bernoulli(0.5), and evaluated $\gamma(\mathbf{S}_s\mathbf{C}[\mathbf{u}])$ for fixed down-sampling rate $s = 4$. Figure 2.4 shows that the obtained γ is very close to the non-asymptotic lower bound (Lemma 3), indicating that this pattern is optimal in a practical sense.



(a) $r = 17, m, n = 50 - 150$

(b) $r = 9, 17, 25, 33, m, n = 128$

Figure 2.4: Reaching theoretical bounds. A random Bernoulli binary image is used to generate the $\mathbf{S}_s\mathbf{C}[\mathbf{u}]$. We set $s = 4$ and estimate γ for different observed image sizes (m, n values) (a) and different kernel sizes (r value) (b). The proximity between the obtained γ and the theoretical bound shows the tightness of the derived γ lower bound.

Concluding Remark. *The mathematical argument and experiments above show that near-optimal γ values are reached with a Bernoulli random noise pattern for reasonable observation, kernel and pattern sizes. Slightly better γ values could be achieved if we allowed the pattern to adapt to the kernel size. This is nevertheless not practical. The payoff would be a negligible improvement of the well-posedness, and the exact* PSF *support size being anyway a priori unknown.*

2.3.3 From continuous patterns to digital patterns

Based on the previous section it comes into view that good PSF estimation patterns are those that produce very contrasted random $\tilde{\mathbf{u}}_D$ sequences. However, we cannot choose directly the values inside the $\mathbf{S}_s\mathbf{C}[\tilde{\mathbf{u}}_D]$ operator. Indeed, the γ value depends on $\tilde{\mathbf{u}}_D$, obtained by sampling on the $s\times$ grid the distorted continuous pattern image.

Consider the set of analogical patterns formed of constant $\mathbf{u}_{ij}$ gray value squares regions,

$$u(\mathbf{x}) = \sum_{i,j} \mathbf{u}_{ij} \mathbf{1}_{\|\mathbf{x}-(i,j)\| \leq \frac{1}{2}}.$$

Since signals in optical systems are non-negative in nature and bounded, we can assume w.l.o.g. that $0 \leq \mathbf{u}_{ij} \leq 1$.

For the mathematical exploration of optimal patterns, we will restrict ourselves to the case where the geometrical transformation D is a zoom-out with factor t^{-1}, $\mathbf{Z}_{t^{-1}}$. This assumption is almost perfectly satisfied if the views of the pattern are taken frontally. Notice that the s-sampling operator can be written as $\mathbf{S}_s = \mathbf{S}_1\mathbf{Z}_s$. Thus,

$$\begin{aligned} \tilde{\mathbf{u}}_D &= \mathbf{S}_s \text{LPF}_s u_D \\ &= \mathbf{S}_1 \text{LPF}_1 \mathbf{Z}_s u_D \\ &= \mathbf{S}_1 \text{LPF}_1 \mathbf{Z}_s \mathbf{Z}_{t^{-1}} u \\ &= \mathbf{c} * \mathbf{u} \end{aligned}$$

where $\mathbf{c}$ is the digital filter

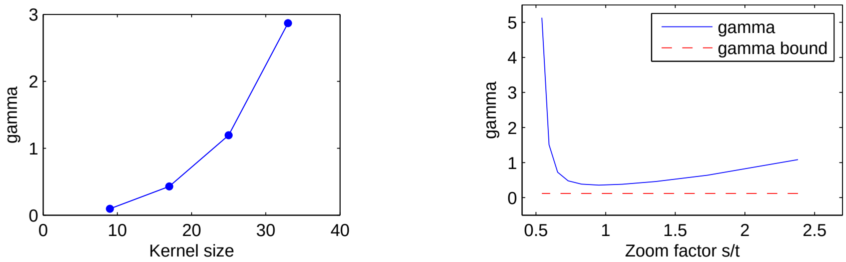
$$c_{i,j} = \int_{-\frac{1}{2}}^{\frac{1}{2}} \int_{-\frac{1}{2}}^{\frac{1}{2}} \operatorname{sinc}\left(\frac{s\zeta}{t} - i\right) \operatorname{sinc}\left(\frac{s\eta}{t} - j\right) d\zeta d\eta.$$

As mentioned earlier the goal is to produce values $(\tilde{\mathbf{u}}_D)_{ij}$ as independent and contrasted as possible. This motivates the following simplification. Suppose that the setup realizes $t = s$.

An ideal unattainable situation would be that the re-sampling operator and the low-pass filter do not produce *inter-symbol interference* (i.e., the discrete filter $\mathbf{c}$ does not change the input signal $\mathbf{u}$). Then each of the square gray values would be equal to the sample after low-pass filtering $\tilde{\mathbf{u}}_D \approx \mathbf{u}_{ij}$. In this particular case we would have a perfect one-to-one correspondence between the gray values of the pattern and the $\tilde{\mathbf{u}}_D$ digital signal which would be a Bernoulli pattern. Due to the constraints on $\mathbf{u}_{ij}$ the best we can do is to choose *iid* random variables $\mathbf{u} \in \{0, 1\}$ with Bernoulli(0.5) distribution. Yet, while this perfect geometric situation is unattainable, the experiments show that γ stays close to its optimal value when s/t is between 0.7 and 2, as it is shown in in Figure 2.5. The resulting $\tilde{\mathbf{u}}_{D_{ij}}$ for distances in a range from $s/t = 1$ produce γ values close to the γ bound for entries in $[0, 1]$.

2.3.4 Comparison of calibration patterns

The γ factor introduced above permits to compare the suitability of different patterns for the PSF estimation problem. Since the noise amplification is governed by the sum of the inverses of the singular values, it is desirable to use patterns that produce singular values that are all as large as possible. For this purpose, and justified by the previous theory, we shall use a binary random pattern. The proposed noise pattern consists of a matrix of 256×256 black and white random



(a) Value of γ as a function of the kernel size (window size 85×85) s=4

(b) Value of γ as a function of window size (kernel Size 17×17) s=4

Figure 2.5: Random Pattern Analysis. Sensitivity of the γ value to the kernel support size (a) and to the t/s zoom factor (b) $s = 4$. The larger the support of the kernel, the noisier the estimation when the gamma value increases with the kernel support size (a). The zoom factor s/t is closely related to the focal distance and to the distance from the camera to the pattern. For example if the distance to the pattern is too small (small s/t value) the pattern will look like a step-edge pattern because of the zoom-in. The corresponding γ value will be higher than the optimal. On the other hand, if the distance to the pattern is large, then the γ value will also be larger than the optimal one, because of the contrast loss due to the zoom-out. In agreement with the theoretical study, the views with zoom factors close to one (i.e., $t \approx s$) produce the best γ values.

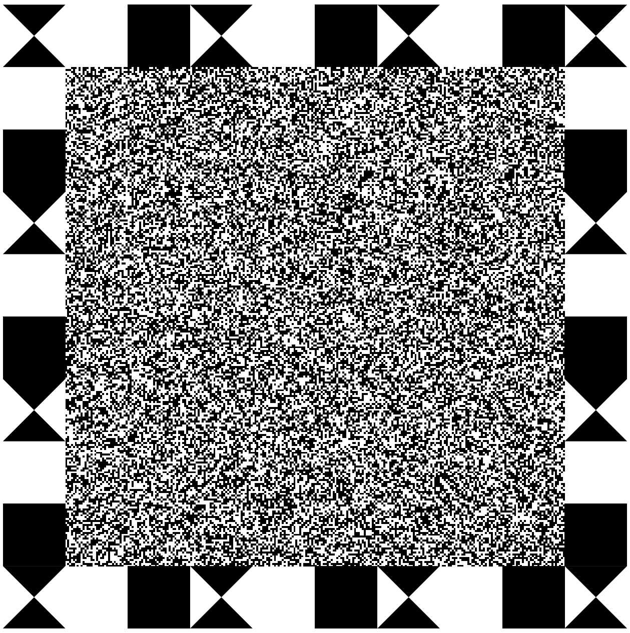
	9×9	17×17	25×25	33×33
Joshi et al. [2008]	99.44	1133.05	6445.87	58419.08
Bernoulli pattern	0.19	0.69	1.54	2.98
Theoretical bound	0.10	0.35	0.70	1.15

Table 2.2: A comparison of pattern realizations through the γ value. The Bernoulli pattern produces significantly smaller γ values than the slanted-edge Joshi et al. [2008] pattern.

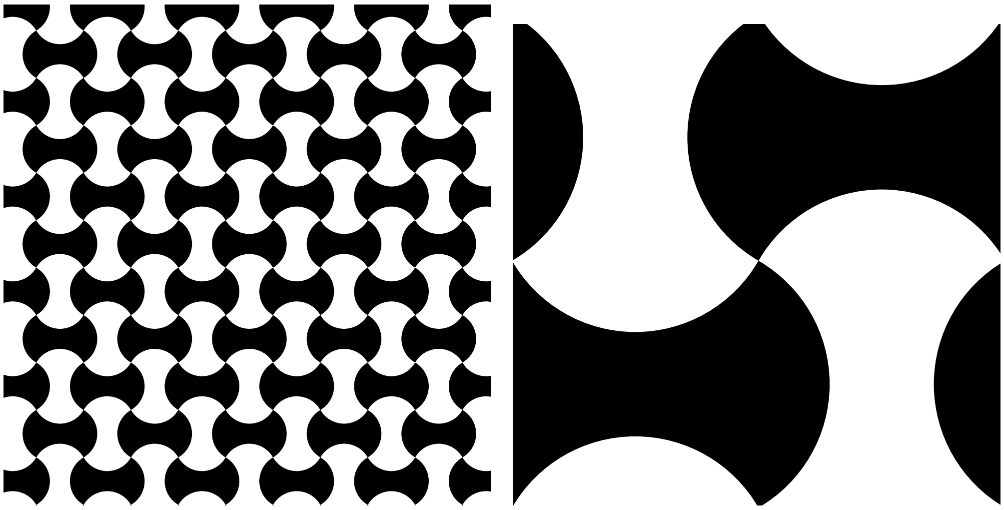
squares generated from an equiprobable Bernoulli distribution. The pattern was printed at a high enough resolution so that artifacts introduced by the printer could be neglected. Several cross marks and white/black flat regions were added, to easily align the acquired image with the pattern, and to correct non-uniform illumination. Fig 2.6 shows the proposed random pattern, compared to a pattern designed by Joshi et al. consisting of $120°$ arc step edges.

Suppose we want to do a $s = 4\times$ PSF estimation. As shown in the previous section, the pattern should be photographed at such a distance that the pattern covers more or less $256/4 \times 256/4$ pixels. In practice, this permits a very local PSF estimation.

Fig. 2.7 shows the eigenvalues of the $\mathbf{S}_s\mathbf{C}[\tilde{\mathbf{u}}_D]$ matrix for $s = 4$, an observed window with size 80×80, and varying kernel sizes, for Joshi et al.'s pattern and for the proposed Bernoulli pattern. The random pattern produces secondary eigenvalues very similar in contrast to the fast decay shown by the eigenvalues of the slanted-edge Joshi pattern. The γ values for the corresponding patterns are shown in Table 2.2. In all cases, the random pattern significantly outperforms the Joshi et. al. pattern. The γ bound value was computed by taking into account the effective observed window size, that is, leaving out the auxiliary region with the checkerboard and flat regions.



(a) random pattern (proposed)



(b) pattern of Joshi et al. [2008] (c) local pattern of Joshi et al. [2008]

Figure 2.6: Calibration patterns for local PSF estimation

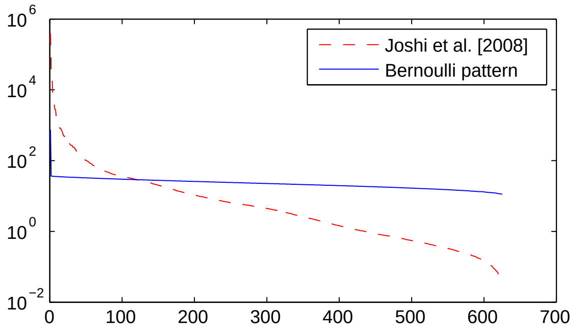


Figure 2.7: Pattern Comparison I. Proposed Random Pattern vs Joshi et al slanted-edge-circles pattern. Observed Window of size 81×81, PSF support size 25×25, $s = 4$. Eigenvalues sorted from highest to lowest. The random pattern produces very similar eigenvalues, and the decay its very slow in comparison to the ones from the Joshi et al. pattern.

2.4 The Complete PSF Estimation Procedure

In this section we describe the steps that lead to a local subpixel PSF estimates. The complete chain is summarized in the block diagram of Fig. 3.2. The next paragraphs present brief summaries for each block. A detailed description is given in Appendix A.

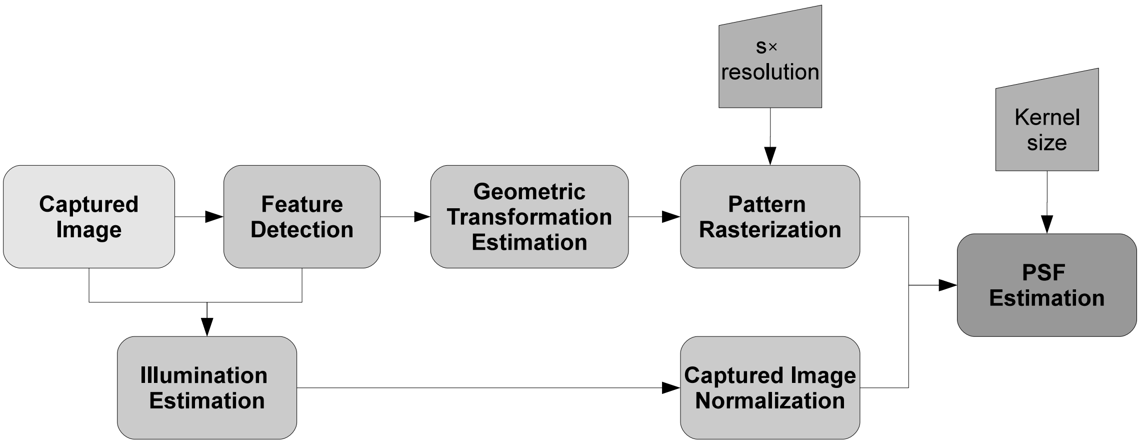


Figure 2.8: Algorithm Description. The captured image is precisely aligned to the analytic pattern through intentionally inserted checkerboard markers. Non-uniform illumination and non-linear camera response function impact - CRF are corrected from the captured image to allow an artifact-free $s\times$ PSF estimation.

Feature detection In order to deal with geometric distortions the ideal pattern and its observation have to be precisely locally aligned. To that purpose checkerboard corners were introduced along the boundary of the noise calibration pattern. Assuming that the PSF is (approximately) symmetric, these x-corners will not suffer from shrinkage. Several methods to detect checkerboard corners have been reported in the Computer Vision literature (e.g. [Harris and Stephens 1988], [Cheng et al. 2005], [Lucchese and Mitra 2002]), ranging from differential operators such as the Harris detector to more specific correlation methods. We used a Harris-Stephens based corner detector implemented by Bouguet [2008], that allows us to iteratively refine the detected corner positions to reach subpixel accuracy.

Geometric transform estimation The estimation of the PSF does not require a decomposition of the distortion into its homography and non-homography parts, as it is done in classical geometric camera calibration [Zhang 2000], where a global radial lens distortion model is usually adopted. In order to avoid that computation and to utilize a more flexible model that may capture local lens distortion, the complete geometric distortion was approximated with thin-plate splines. While thin-plate splines were originally conceived as an exact interpolation method [Bookstein 1989] they can be easily extended to the approximation problem [Sprengel et al. 1996]. The mapping from the non-distorted to the distorted space is estimated from the detected corners $\{\tilde{p}_i\}$, and their correspondences in the ideal pattern $\{P_i\}$, whose coordinates are perfectly known. Accurate geometric transform estimation is essential for good performance. Although there are no "control points" inside the random pattern (only on borders as depicted in Fig. 2.6(a)), as the pattern is designed to cover about 100×100 pixels in an image, the local geometrical distortion inside such a small region will be practically affine. Hence, by using thin-plate splines we can achieve the necessary registration accuracy.

Illumination estimation and normalization In order to match the gray levels in the sharp pattern to those in the observed image, black and white square flat regions were included along the boundary of the noise pattern. These regions permit to estimate the mapping between black and white colors and the corresponding observed gray level values. The presence of these constant regions all around the pattern permit to estimate a black (white) image that models the black (white) intensity level at each pixel. These light images have been modeled by second order polynomials whose coefficients are estimated by least squares from the known pairs (value, position). In continuation each pixel value in the observed image is linearly rescaled within the range $[0, 1]$, by considering the respective estimated black and white values.

CRF estimation $g(\cdot)$ Once the nonuniform illumination has been compensated, the camera response function can finally be estimated and the non-linear response of the sensors corrected. Since we are working with the RAW data and out of the saturation region of the sensors, the sensor response should be almost linear. Hence, for simplicity we model the camera response function as a polynomial of order no larger than 2. The estimation and correction procedure is based on a strong property of our pattern: the white noise pattern 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 is defined as a parabolic function $u \mapsto \alpha u^2 + (1 - \alpha)u$ where α is chosen so that the mean of the pattern after the correction is $1/2$.

Pattern rasterization In order to generate the samples $\tilde{\mathbf{u}}_D$ from the ideal continuous pattern image u, we need to sample this image at the desired s resolution after deforming it by the estimated geometric transformation. For that purpose the distorted continuous pattern u_D must be low pass filtered to be band-limited in $[s\pi, s\pi]$. (Remember that the camera resolution is $1\times$. Thus the digital pattern has an $s\times$ over-sampling). The procedure is:

1. The continuous pattern u is sampled at a very high resolution. From the vectorial description of the pattern a digital image is generated (this procedure is called rasterization) by

replacing each one of the flat squares by a 4×4 block of pixels with the same gray value. The re-sampling starts from these samples $\mathbf{u}$ instead of the continuous pattern;

2. Frequencies higher than $s\pi$ are cut off from the digital pattern $\mathbf{u}$ to get $\tilde{\mathbf{u}}$;

3. By help of the previously computed geometric distortion the filtered pattern $\tilde{\mathbf{u}}$ is bi-cubically interpolated at the desired resolution $s\times$ $\tilde{\mathbf{u}}_D$.

Numerical methods for PSF estimation We have seen that light diffraction, optical low pas filtering, and sensor light integration all produce non-negative kernels. Thus the estimated PSF must be non-negative. We can therefore constrain the solution to be nonnegative, thus reducing the space of solutions. Section 2.5, Fig. 2.18 shows that not imposing this non-negativity assumption yields essentially the same results, which in fact verifies the correctness of the proposed image formation model. Hence, we can opt to solve a non-negative least squares, or to simply solve a least squares problem and then threshold the solution to eliminate very little components.

Suppose that the local grid pattern observation $\mathbf{v}$ has size $m \times n$ and that we want to estimate a PSF at $s\times$ subpixel resolution. Also suppose that the estimated support of the PSF is inside a $r \times r$ image. The matrix $\mathbf{S}_s\mathbf{C}[\tilde{\mathbf{u}}_D]$ corresponding to the s-down-sampling of the convolution with the distorted calibration pattern, has size $mn \times r^2$. Thus, the problem to be solved can be formally written as

$$\arg\min_{\mathbf{h}} \|\mathbf{S}_s\mathbf{C}[\tilde{\mathbf{u}}_D]\mathbf{h} - \mathbf{v}\|^2 \qquad \text{subject to } \mathbf{h}_i \geq 0,\ i = 1, \dots, r^2. \tag{P$'$}$$

Problem (P$'$) can be solved using standard convex optimization solvers such as CVX [Grant and Boyd 2012]. A simpler Newton interior point algorithm proposed in [Portugal et al. 1994] was used and always converged rapidly.

2.5 Experimental Results

This section is dedicated to the evaluation of the proposed non-blind subpixel PSF estimation method, and to the comparison of its performance with two state of the art proposed approaches. A complete algorithmic description, an online demo facility and a source code can be found in our IPOL publication Delbracio et al. [2012b].

We selected a method recently reported in the literature by Joshi et al. [2008], and a MTF commercial software, Imatest [LLC 2010]. Since we do not have real camera ground truth for the PSF, the performance evaluation was first carried out on simulated data. A real PSF estimation on real cameras was in continuation tried under varying acquisition conditions. Particular attention was paid to the aliasing effect caused by sampling under the Nyquist frequency.

2.5.1 Simulations for objective evaluations

The simulation of the camera acquisition process was as follows. The grid pattern was rasterized at a very high resolution (i.e., $8\times$), convolved with a PSF like kernel (in this case a Gaussian isotropic kernel), and down-sampled to get the observed digital image at the camera resolution (i.e., $1\times$). The kernel was chosen so that the low resolution image presented aliasing artifacts. We also added white Gaussian noise of standard deviation $\sigma = 0.02$. We compared the performance of the proposed approach to that of Joshi et al. using their calibration pattern and our implementation of their approach, with different regularization levels. A $4\times$ kernel was estimated for both algorithms from the observed window of size 110×110 pixels.

Figure 2.9 shows the results for $4\times$ PSF estimation from the simulated observation. Solutions with the Joshi et al. method with three levels of regularization are presented, along with the proposed approach (which is regularization-free). In this experiment the proposed method significantly outperforms Joshi et al.'s algorithm, achieving a much less noisy estimation. Joshi's algorithm needs a strong regularization to stabilize the estimation and to avoid an amplification of high frequency noise. Consequently, its estimation tends to penalize high frequency components and to produce a biased kernel with amplified lower frequency components. See caption for details.

2.5.2 Experiments with real camera images

In this section we present several 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. The impact of the CRF estimation/correction was evaluated, and the PSF estimates obtained for the four color channels in the Bayer pattern compared. Variations of the kernel estimates depending on their location in the image were also explored. This was followed by an evaluation of the stability of the estimation procedure, and of the influence of the kernel support size. Finally the results were again compared with the Joshi et al. algorithm and with Imatest, applied to real cameras.

Different apertures The estimation was conducted using the proposed random pattern captured at five different apertures. For each acquisition, a $4\times$ PSF estimation for one of the green channels (half of the green pixels of the Bayer matrix) was performed. Results are shown in Figure 2.10. The estimations were performed at the image center from a window of size 90×90 pixels. Notice that kernels at apertures f/32 and f/16 are significantly larger than the rest, as predicted by diffraction theory (see caption for details). An example of the acquired blurry image is shown in Figure 2.13.

Fig. 2.12 shows the diffraction-limited MTF for a circular f/5.6 aperture and green monochromatic light (See the end of Sec. 2.2.1). The estimated response for our camera-lens system at aperture f/5.6 and for the green channel is under the ideal diffraction-limited response. This can be a consequence of the light integration in the sensor array but also of the optical low pass filter specifically included to avoid aliasing.

Estimation of camera response function This experiment evaluates the impact of the non-linearity of the camera sensors response. To conduct this experiment, the camera response curve was computed using a specially designed pattern for CRF estimation. In order to assess the impact of the CRF on the PSF estimation, the observed image was corrected using the special purpose CRF estimate, to compare the results that yield the PSF estimation algorithm.

Figure 2.14(a) compares the CRF estimated using the special purpose pattern with the CRF estimate embedded in the proposed PSF estimation algorithm. Notice that both estimates are hardly non-linear and extremely close to each other, so the PSF estimation algorithm seems to be capable of giving a reasonable CRF estimation. Figure 2.14(b) shows the MTFs obtained under four different situations:

- estimation with embedded CRF correction from the raw observed values (`psf-crf`).

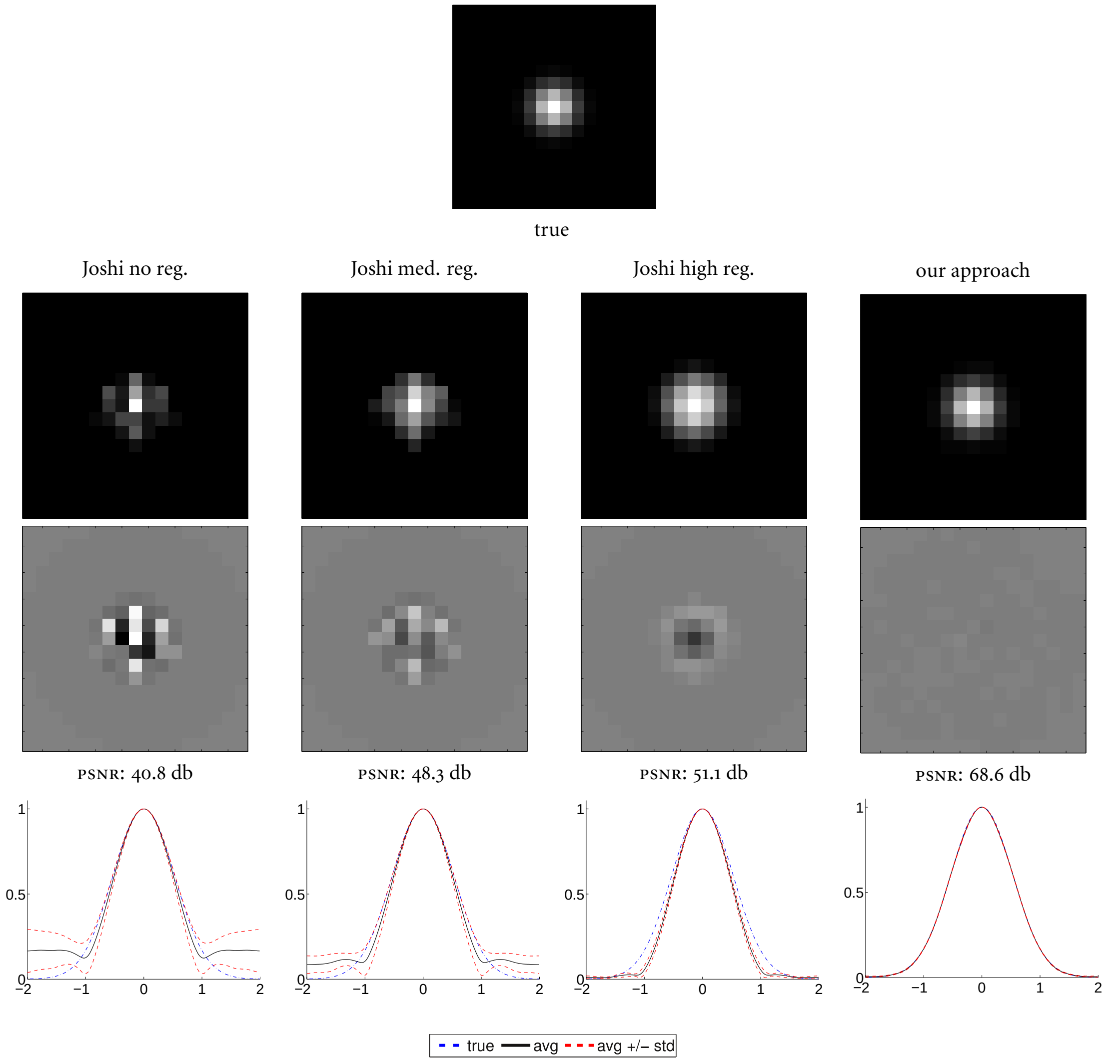
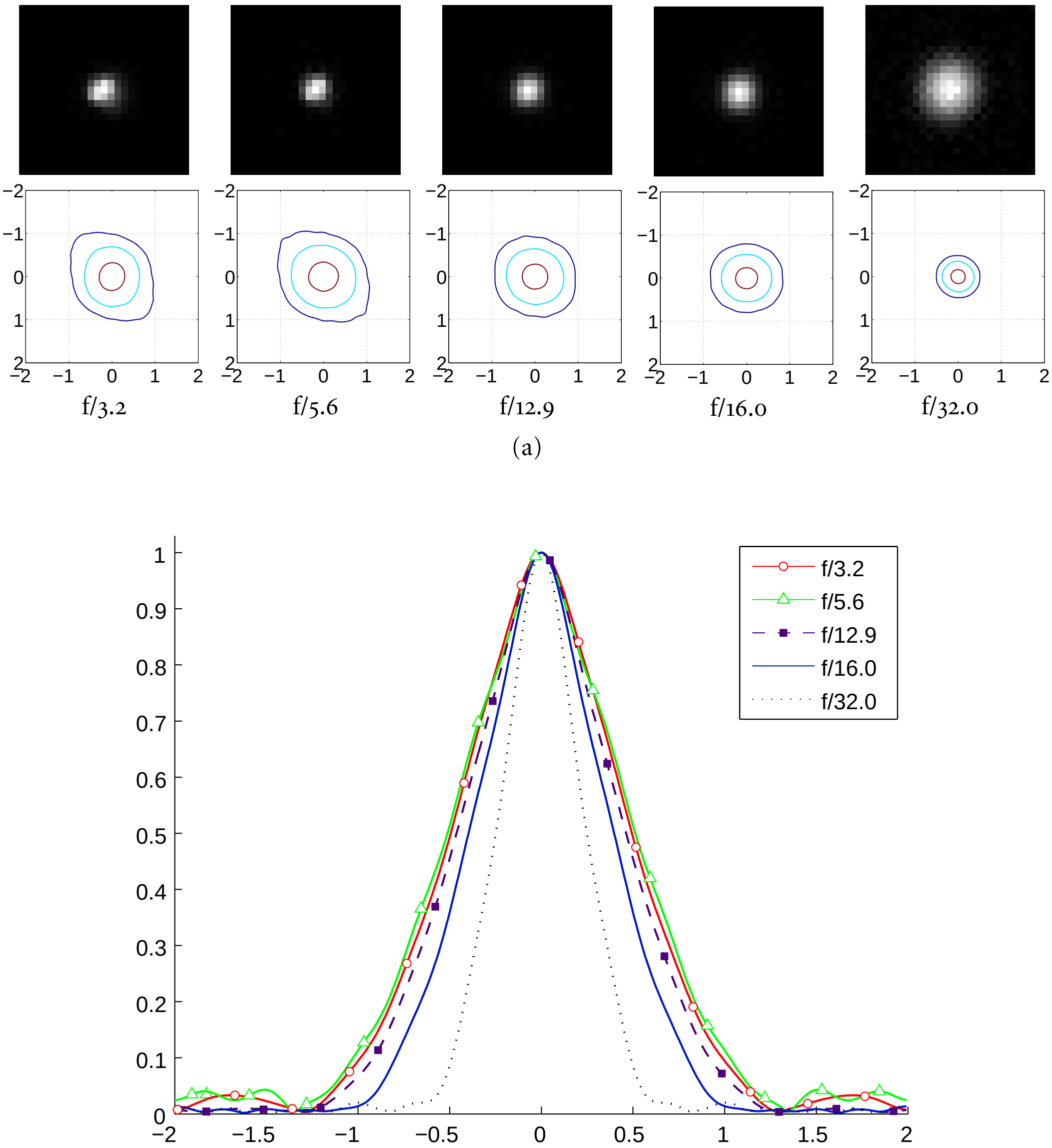


Figure 2.9: Synthetic example I. Performance comparison for simulated data. A $4\times$ kernel is estimated using the Joshi et al. algorithm, with varying regularization level, and the proposed approach. The observed window has 110×110 pixels. The top row shows the kernel estimation and the middle row the difference image between the estimation and ground truth for one of the realizations. The proposed method significantly outperforms the Joshi et al. algorithm, achieving a much less noisier estimation as shown by the difference images and by the peak signal to noise ratios. The bottom row shows central horizontal profiles for all the estimated MTFs (0.5 is the Nyquist frequency). Notice that in the Joshi et al. method the estimation is unstable. The estimates show extremely noisy components for frequencies higher than the sampling frequency, when the amount of kernel regularization is too small. On the other hand, if a strong regularization is imposed, the penalization of the kernel gradient adopted by Joshi et al. tends to produce kernels with under-estimated high frequency components. The method proposed here does not rely on a regularization and produces nonetheless noiseless and unbiased results.



(a)

(b)

Figure 2.10: Different apertures. Taken at different apertures, green channel g1, 100 ISO, 50mm. All estimated $4\times$ kernels are quite smooth. Fig. (a): The top and bottom rows show respectively the estimated PSFs and a few level lines of the corresponding MTFs that prove that the kernels are not exactly axis-symmetric. The kernels at apertures f/32 and f/16 are considerably larger than the rest in agreement with diffraction theory. This phenomenon also stands out in the modulus of the estimated PSF spectra, which also shows that the PSFs/MTFs are not axis symmetric. Figure (b): Vertical cuts of the spectrum modulus. The camera seems to have the sharpest response from apertures f/3.2 to f/12.9. At apertures f/32 and f/16 the camera cuts high frequencies significantly more than the rest, as predicted by diffraction theory. Notice that in all cases, except at aperture f/32, the MTF at the Nyquist frequency ($f = 0.5$) is significantly greater than zero. Hence, the camera introduces aliasing.

Figure 2.11: Real camera example. Taken at aperture f/5.6. An example image, to show how local the PSF estimation is (left), and a zoom of the observed window of size 110×110 pixels (right).

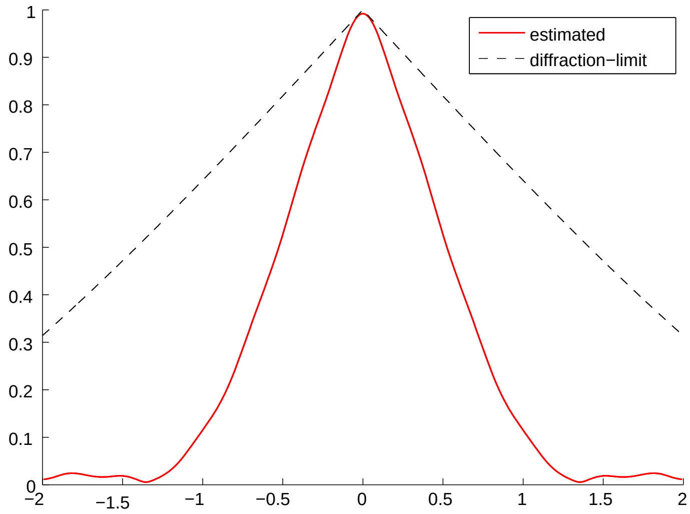


Figure 2.12: Diffraction-limited System. Theoretical diffraction MTF for monochromatic green light with circular f/5.6 aperture and the estimation for the green channel at the same aperture. The estimated response for our camera-lens system is under the ideal diffraction-limited response. This can be consequence of the light integration in the sensor array, but also of the optical low pass filter specifically included to avoid aliasing.

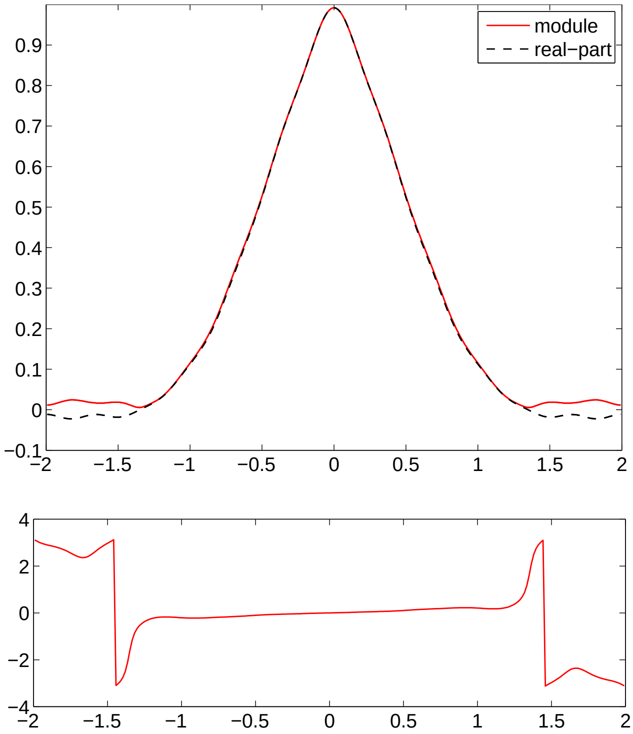
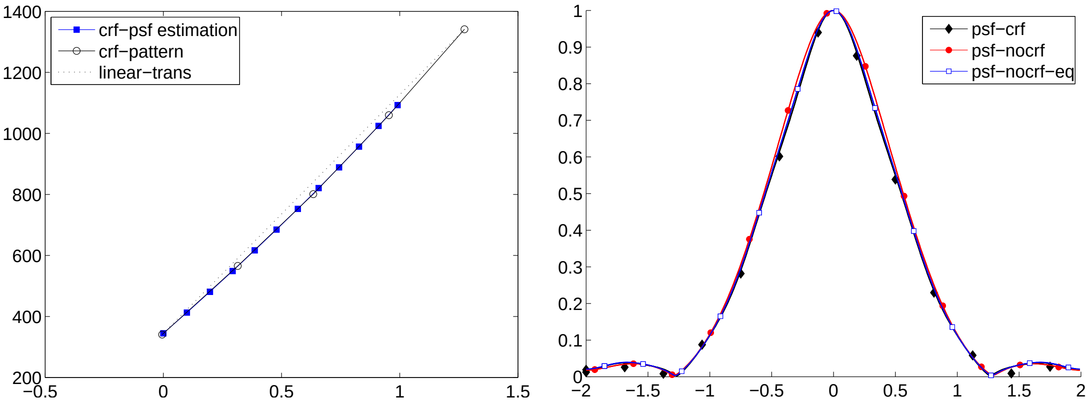


Figure 2.13: OTF phase. Estimation done for the green channel g1, f/5.6 at the center of the sensor array. The Figure on the top shows the modulus of a horizontal profile of the optical transfer function - OTF and its real component. Both curves coincide, implying that the OTF is real and thus the PSF is symmetric. This is also seen in the bottom figure that shows that the OTF phase is 0 or π.

- estimation without any CRF correction from the raw observed values (`psf-nocrf`).
- estimation without embedded CRF correction from the adjusted values after correction *via* the special purpose CRF estimate (`psf-nocrf-eq`).

In all cases, the estimation yielded very similar results.



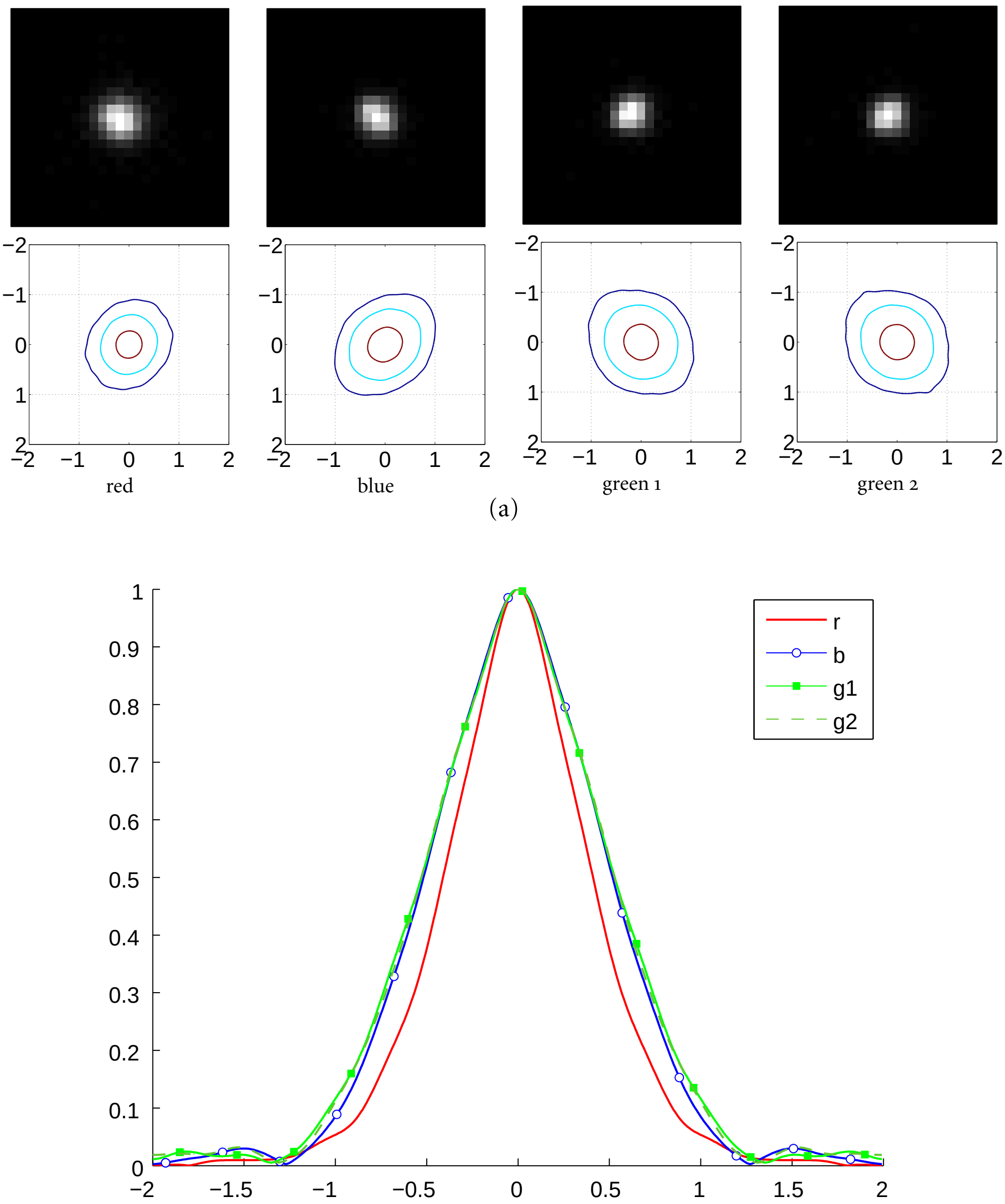
(a) CRF estimates (b) Vertical profile of MTFs

Figure 2.14: Dependence on the CRF correction, for a $4\times$PSF estimation of the green channel, at aperture f/5.6. Figure (a): CRF estimates obtained with the estimation embedded in the proposed PSF estimation algorithm (`crf-psf estimation`), and with the one generated independently from a special purpose CRF calibration pattern (`crf-pattern`). Both estimates are very similar and hardly non-linear. Figure (b): vertical profile of MTFs. The estimates from the raw gray values with and without CRF estimation/compensation (`psf-crf` and `psf-nocrf`, resp.) gave very similar results. After compensation of the gray values using an external special purpose estimation of the CRF, the PSF estimation procedure (`psf-nocrf-eq`) also led to very similar results.

Color estimation The goal of this experiment is to compare the PSF estimates for all four channels from the Bayer RAW camera output (two greens, red and blue). The estimation was performed using the random pattern captured at apertures f/5.6. The results for the $4\times$ PSF estimation located in the image center are shown in Fig. 2.15. It is easily seen that the red PSF is larger than the green and the blue one (i.e., produces blurry images). This is reasonable, since the wavelengths associated to red are smaller than the rest. Hence the red diffraction kernel will be larger than the green and blue kernels for the same camera configuration. The differences between the shapes of the red/blue and green PSF spectra can be explained by the sensor shape. If we accept that the sensor active zone is L-shaped, then by the red/blue sensors in the Bayer pattern will have the same sensor term MTF and will be rotated $45°$ with respect to the green channels.

Location Figure 2.16 displays the $4\times$ PSF estimates for one of the green channels, at different image locations, for f/5.6. 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.

Stability of the estimation procedure A set of thirteen images of the noise calibration pattern were acquired with exactly the same camera configuration (f/5.6), from similar viewpoints.



(a)

(b) Horizontal profile (MTF)

Figure 2.15: Different color channels. $4\times$ PSF estimation for the four Bayer pattern channels (two greens, red and blue) from a camera RAW output. Top row: PSF estimation. Middle row: the corresponding Fourier spectrum moduli. Bottom row: MTF horizontal and vertical profiles. The estimation was performed using the random pattern captured at aperture f/5.6. The red PSF is larger than the green and blue ones. Since the wavelengths associated to red are smaller than the rest, the diffraction components for the red channel will be larger than those for green and blue for the same camera configuration. Also note the differences between the shape of the red/blue and green PSF spectra (bottom row). Red and blue MTF seem to be $45°$ rotated with respect to the green ones. This symmetrical behavior is plausible for an L-shaped active zone sensor array.

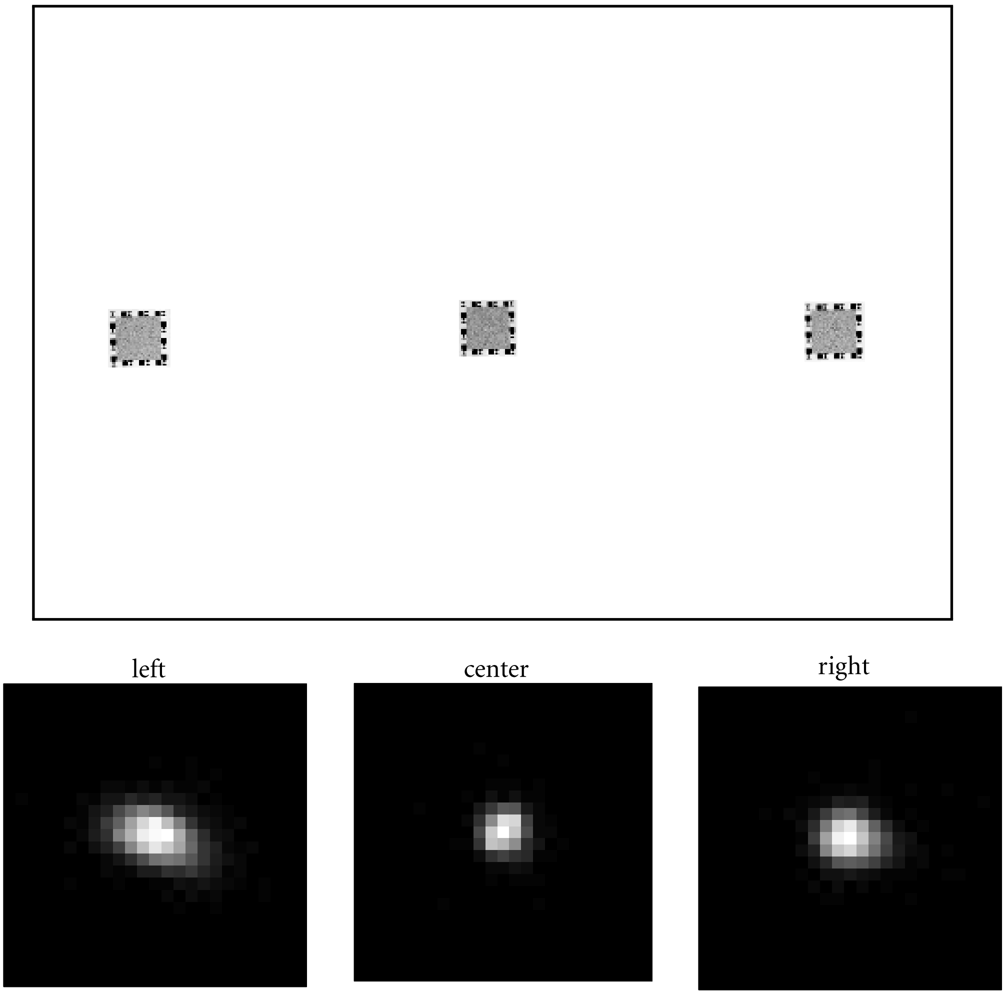
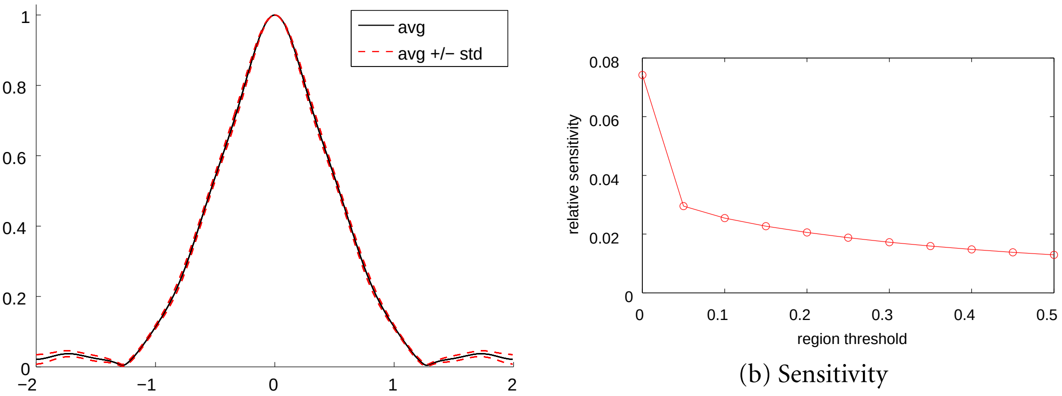


Figure 2.16: Different locations. Taken at f/5.6 for one of the green channels. The PSFs estimated far from image center are larger and more asymmetrical than the one estimated at the center. This is certainly due to lens aberrations, which are more significant near the image borders.

For each acquisition, the $4\times$ PSF of one of the green channels at the image center was estimated. Figure 2.17 shows the average MTF vertical profile, and its standard deviation band. It is clear from the small value of the standard deviation that the estimation method is highly stable, in agreement with the fact that the corresponding linear system to be inverted is very well-posed. More details are given in Fig. 2.17 caption.



(a) MTF vertical profile

(b) Sensitivity

Figure 2.17: Stability of the estimation. Average and standard deviation statistics were generated from 13 estimations computed at f/5.6 (for one of the green channels). The small standard deviation in the vertical profile of the Fourier spectrum modulus is shown in (a). The relative MTF sensitivity vs region threshold is shown in (b). We define the relative MTF sensitivity in a region Ω as: $s(\Omega) = \text{mean(std(mtf))/mean(mtf)}$ where the mean values are computed inside the region Ω. In this case we construct $\Omega(\text{threshold}) = \{x : \text{mtf}(x) \geq \text{threshold}\}$. The relative sensitivity in the whole spectrum does not exceed 0.08 and what is more if the MTF values smaller than 5% are not considered, then the relative sensitivity is less than 3%. The small standard deviation and sensitivity demonstrate the method stability.

Support We can consider that the proposed approach has only one main parameter: the kernel support size. The choice of this size implies a trade-off between the model validity and the feasibility of the estimation. On the one hand, if the support is too large the kernel estimation will be very noisy, since the γ factor increases with the support size. On the other hand, if the kernel support is too small the considered image formation model will not be accurate.

Fig. 2.19 shows the $4\times$ PSF estimation for various kernel support sizes. All estimations for the supports 17×17, 25×25 and 33×33 turn out to be very close to each other. Nevertheless, the 9×9 kernel support does not seem to be large enough to correctly model the PSF. Hence, as soon as the support size exceeds such a lower bound, the proposed algorithm does not appear to be sensitive to this parameter.

Comparison of several methods This section ends up with a comparison between the Joshi et al. method, Imatest, and the proposed approach to non-blind subpixel PSF estimation [LLC 2010]. Imatest is a commercial MTF estimation software. The Imatest estimation is performed from a slanted-edge image and only gives an estimate of the MTF at the direction orthogonal to the slanted-edge. The estimation was conducted with images taken at aperture f/5.6 with patterns located at the center of the image.

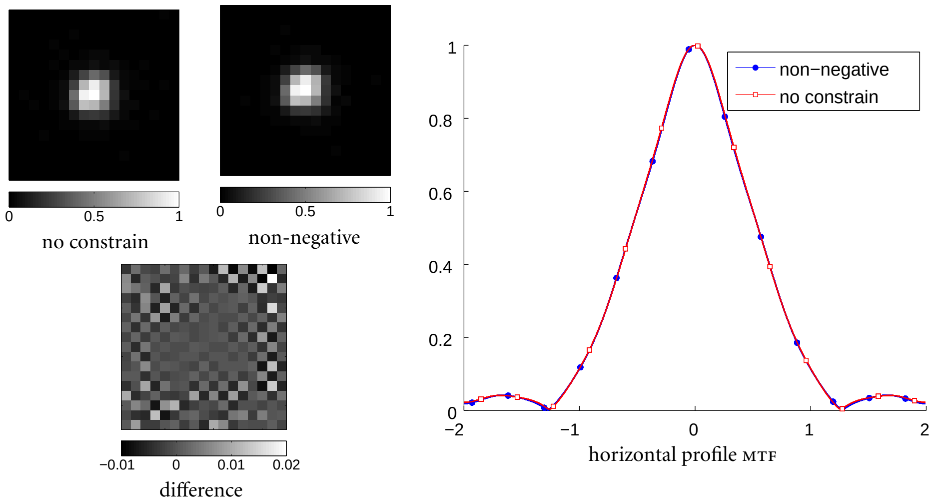


Figure 2.18: Non-negative Constraint. This experiment analyzes how the PSF estimation changes by not assuming the non-negative hypothesis. On the left we show both estimations: the no-constrained and the non-negative $4\times$ PSF for the green channel, f/5.6 at the center of the image. Since there is no structure in the image produced by subtracting both estimations and since the relation between the energy of the image difference and the energy of the non-negative estimation is 0.001, we can conclude that both estimations are extremely close. This is confirmed by observing in the left figure a horizontal profile of the MTF for both estimations.

Figure 3.10 shows the horizontal MTF profiles obtained with the Joshi et al. method using various regularization levels, with Imatest, and with the proposed approach for one of the green channels. In the low frequencies Joshi and the proposed approach yield very similar results. However, for higher frequencies the Joshi et al. results vary strongly with the regularization level. The Imatest estimate is quite noisy and does not resolve frequencies above twice the sampling rate. The proposed random pattern algorithm generates much more information than the typical slanted-edge MTF calibration.

2.6 Discussion

The work presented in this chapter is an attempt to define an optimal non-blind subpixel PSF estimation method from a single *aliased* image. The method is successful, but its setup is tight. The pattern must be large enough (some 70cm in our experiments), printed with good quality ink. The random squares must be large enough to avoid any ink soaking bias, and a good quality print is recommended. The mathematical analysis demonstrated that a Bernoulli pattern is nearly optimal in terms of well-conditioning of the matrix to inverse. The pattern was therefore placed in an approximately frontal position. The photographs were taken at the right distance to ensure that the camera sampling grid and the pattern grid had similar meshes. These position requirements are not strict, though, the experiments showing only a slow degradation of the results when the distance varies around the optimal position. The method is also very strict in the precautions to compensate for the variations in illumination and to estimate the exact deformation between the ideal pattern and the observed one.

Nevertheless, the pay off of this careful procedure is high. The method delivers a very accurate

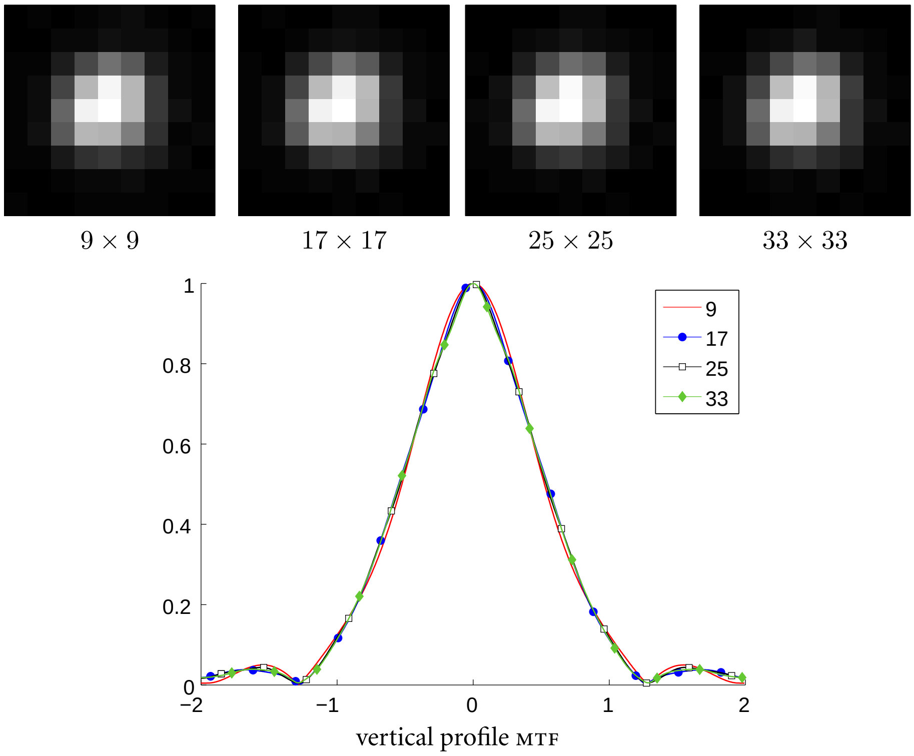

9×9 17×17 25×25 33×33

Figure 2.19: Changing support size. This experiment analyzes how the PSF estimation changes with the desired PSF support size. Several PSF estimations for various kernels support sizes (left). Only the central 9×9 regions are shown. All the estimates are very close, specially 17×17, 25×25 and 33×33. However, the 9×9 kernel support seems to be hardly sufficient for correctly modeling the PSF, as indicated by the MTF vertical profiles on the right. The proposed algorithm does not appear to be sensitive to this parameter as soon as the kernel support exceeds this minimal size .

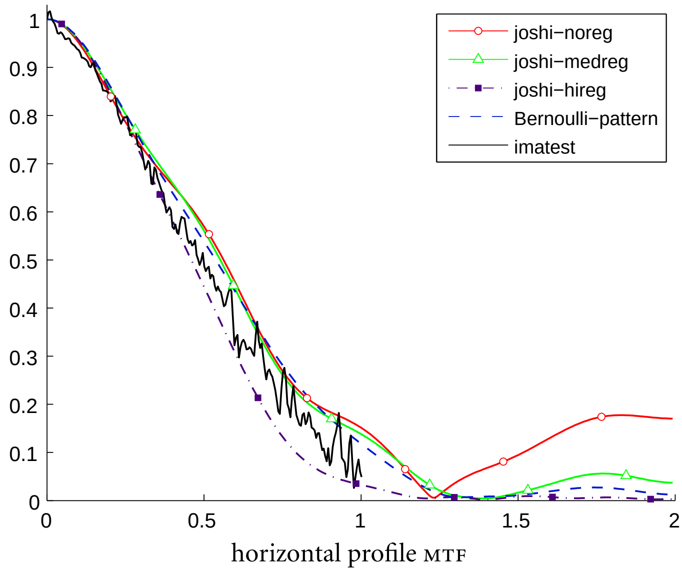


Figure 2.20: Comparison of PSF/MTF estimation methods applied to a real camera. Our implementation of Joshi et al. [2008] PSF estimation algorithm, the Imatest commercial software and the proposed random patter algorithm. All estimations are done at the center of the image with a camera at aperture f/5.6 for one of the green channels. On the low frequencies all algorithms gave very similar estimations, while on the higher frequencies the Joshi et al. estimation depends on the regularization level. Although we did our best to get a noise free MTF estimation from the Imatest software, the final estimation is quite noisy. The Imatest estimation is done from a slanted-edge image and only gives an estimation for the MTF at the slanted-edge orthogonal direction.

estimate of the PSF, as amply shown in the various comparative experiments, with quite stringent accuracy levels (relative error in the order of 2% to 5%). It remains to wonder why the former methods added regularizing terms or *a priori* models if these were not needed. Yet, the numerical experiments have confirmed that the inverse estimation problem is indeed ill-posed with slanted edge patterns, which accounts for the necessity of regularization terms for such patterns. Although random noise patterns have been widely used in the past, up to our knowledge no regularization-free subpixel PSF estimation scheme had been previously proposed. For these previous methods with noise patterns, the lack of a careful correction for all perturbations may explain the need for a regularization or an *a priori* model. The experiments here have confirmed that for typical DSLR cameras, each color channel is under-sampled with respect to the ideal Nyquist rate given by the PSF, by a factor of 2 or even 4. This fact was confirmed, even with DSLR models including an optical anti-aliasing filter on the sensor. This more than justifies *a posteriori* the need of a subpixel estimation procedure. As usual, a locality-accuracy trade-off had to be resolved. The locality of the order of a few hundred pixels can be achieved under common noise conditions.

Of course a wholesome local camera calibration remains a heavy procedure. According to the above setting, some 100 snapshots of the pattern are needed to cover the whole image domain to get an accurate enough PSF estimate everywhere. Indeed, the experiments show that this kernel varies significantly, particularly near the image boundaries. A possible solution to avoid these many photographs would be to print a very large random pattern covering a whole wall, that would cover the whole visual field of the camera. While this is not easy to implement, it is indeed doable in lab conditions.

3 PSF Estimation from Two Photographs at Different Distances

In most digital cameras, and even in high-end digital single lens reflex cameras, the acquired images are sampled at rates below the Nyquist critical rate, causing aliasing effects. This chapter introduces an 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 herein proves that the camera PSF can be derived from these two images, under reasonable conditions. Mathematical proofs supplemented by 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.

3.1 Introduction

Light diffraction, lens aberrations, sensor averaging and anti-aliasing filters are some of the inherent camera factors that unavoidably introduce blur in photographs. 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), which corresponds to the space variant impulse response of the whole camera, including the sensor, before the final sampling.

The area enclosed by the first zero crossing of the PSF, usually called Airy pattern, is arguably the most reasonable characterization of the optical system resolution. Top camera/lens manufacturers use charts based on the PSF Fourier spectrum modulus (the modulated transfer function, MTF) to describe their products. But accurate knowledge of the PSF is not limited to quality assessment of optical devices, and it proves to be extremely useful or even necessary for several image processing tasks such as deblurring [Ng et al. 2000], superresolution [Park et al. 2003; Robinson and Milanfar 2006] or shape from defocus [Chaudhuri and Rajagopalan 1999].

In most typical digital cameras, both compact and high-end DSLRs, images are sampled at frequencies below the Nyquist critical rate. Consequently, only aliased versions of the camera PSF can be directly observed. Yet, to fully characterize the PSF, it is necessary to recover it at a subpixel resolution.

PSF estimation methods can be classified as blind or non-blind, depending on whether they use or not snapshots of a specially designed calibration pattern. Blind approaches try to estimate the PSF from photographs of an unknown scene. They do assume, however, that the scene involved in the estimation follows some statistical model of sharp images or include a significant amount of geometric cues such as sharp edges. Most of these PSF approaches attempt to detect edges, which are modeled as pure step-edge functions convolved with the PSF kernel [Chalmond 1991; Luxen and Förstner 2002; Capel 2004; Smith 2006]. In this setting, the 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. Therefore, such blind estimation techniques do not lead to accurate PSF estimates and are consequently constrained to image restoration problems, where precision is not the main objective. For this reason, accurate PSF estimation procedures rely on the use of specially designed calibration patterns. A local kernel estimation is performed by comparing the ideal calibration pattern to its photographs.

As we have presented in the previous chapter, several patterns have been used for PSF estimation, ranging from pin-hole, slanted-edge [ISO 2000; Reichenbach et al. 1991; Zandhuis et al. 1997; Claxton and Staunton 2008], or arc-step-edge patterns Joshi et al. [2008] to random noise images [Daniels et al. 1995; Levy et al. 1999; Backman et al. 2003, 2004; Brauers et al. 2010]. Even non-blind subpixel PSF estimation methods reported in the literature led to ill-posed inverse problems. The inversion required the imposition of simple PSF parametric models or other regularity or symmetry priors. In Chapter 2 we have shown that such *a priori* assumptions on the PSF are actually unnecessary and jeopardize the estimation accuracy. More precisely, by carefully modeling the image acquisition system, a calibration pattern made of a white Bernoulli noise realization is nearly optimal in terms of well-conditioning of the problem. This procedure leads to very accurate regularization-free subpixel PSF estimation.

The purpose of the present work is to explore the feasibility of obtaining accurate PSF estimates, while avoiding the explicit use of a calibration pattern. The motivation comes from the fact that, although very precise, the use of a calibration pattern can be sometimes tedious and impractical: these approaches rely on a careful setup, and the calibration grid has to be properly assembled, whereby a good quality print is essential.

We show that, instead of using a photograph of a known calibration pattern, two photographs of the same scene acquired at different distances with fixed camera configuration are enough to recover a regularization-free subpixel PSF. The proposed acquisition procedure is simple and handy in comparison to a non-blind approach. Experimental evidence will show that the resulting estimates do not exhibit any significant accuracy loss compared to their best non-blind competitors. The choice of the photographed scene is important but not critical. For a wide range of everyday textured scenes, the acquired image pairs lead to well posed inversions and highly accurate results. The proposed method can be used with the previously introduced Bernoulli pattern as well, the difference being that in that case the quality requirement for the printed pattern are no longer stringent.

This chapter is written with a dual public in mind: mathematicians and/or image processing specialists. We have tried to define accurately all mathematical objects necessary to deal rigorously with image formation. An accurate formalism is needed to justify the somewhat intricate interlacement of sampling and convolution operations. This forces one to check on the compatibility of all function or distribution spaces to which the objects belong and to verify that the formulas are mathematically consistent. Nevertheless, the application-oriented reader can skip the proofs and the functional space details at a first reading, and simply focus on the standard image processing formalism and algorithms. Most proofs are placed at the end of the chapter. A glossary is appended to display all notation in a single place.

The chapter is organized as follows: Section 3.2 presents a mathematical model of the digital image acquisition system. This model is used in Section 3.3, where it is shown that the camera PSF can be recovered from a pair of unknown scaled images. We define the notion of blur between such a pair of images, and we propose a method to perform its estimation. Then we prove that the camera PSF can be recovered from this inter-image blur. Section 3.4 presents an algorithm that implements the complete PSF estimation procedure described in Section 3.3. In Section 3.5 we discuss a series of experiments on real and simulated images. Finally, Section 3.6 closes with a brief recapitulation and conclusions. The details of the adopted notation and the complete mathematical proofs are presented in Appendices 3.A and 3.B respectively.

3.2 Image Formation Model

3.2.1 Generalized digital pin-hole camera

An accurate estimation of the PSF requires a proper modeling of the digital image formation process. The geometric component of this process is most often modeled in computer vision by a pin-hole camera. An *ideal pin-hole camera* with focal length f, shooting at a planar scene u from a distance d and at fronto-parallel pose, will produce an image $w(\mathbf{x}) = u(\lambda\mathbf{x})$ which is just an homothecy of scale factor $\lambda = \frac{d}{f}$ of the original planar scene u.

If the pose is not perfectly fronto-parallel or the pin-hole camera presents non-canonical internal calibration parameters, w and u are related by a planar homography D, i.e., $w = u \circ D$. In a more accurate camera model the distortion D takes the form of a more general (but regular) diffeomorphism. This is required when the scene is a regular close-to-planar surface (as it is assumed here) or when the geometric distortion due to the optical system is taken into account as suggested in Chapter 2.

For the purpose of PSF estimation this simple model needs to be augmented with an accurate radiometric component, comprising at least the following elements.

Blurring

The PSF kernel h models blur due to intrinsic camera characteristics, such as diffraction when light goes through a finite aperture, light averaging within the sensor and lens aberration. Other blur sources such as motion, atmospheric turbulence or defocus blur, that may change from one snapshot to another, will be minimized by the experimental procedure, and it is not the goal of the present work to estimate them. Another implicit assumption that is usually made is that as long as the camera is in focus, the PSF is independent of the focus position, i.e., the relative distance between the sensor array and the lens system. Therefore *in focus* images captured with the same camera configuration are affected by the same PSF.

The diffraction kernel is determined by the shape and size of the aperture, the focal length, and the wavelength of the considered monochromatic light. Under the Fraunhofer far-field approximation, for incoherent light this kernel is the squared Fourier transform modulus of the camera's aperture indicator function [Goodman 1996]. It follows that the PSF diffraction kernel is always non-negative and band-limited.

Besides the kernel due to diffraction, other sources of blur inherent to the optical system are present in real cameras. These are mainly optical aberrations, and anti-aliasing filters (which reduce aliasing but do not completely cancel it) introduced in the system prior to sampling [Williams and Becklund 2002; Zhao et al. 2006]. The sampling process also introduces blur. Indeed, each photo-sensor in the rectangular sampling grid integrates the light arriving at a particular exposure time. This corresponds to a convolution with the indicator function of the photo-sensor active area. To sum up, *the unknown* PSF results basically from the convolution of three non-negative kernels (diffraction, aberrations and anti-aliasing filters, and sensor averaging), one of them being band-limited. No parametrical model on the PSF will be adopted here. Nonetheless the physical modeling justifies our assumption that the PSF is *band-limited and non-negative.*

Sampling

We model the continuous to digital conversion at the image plane by the introduction of an ideal *sampling* operator $\mathbf{S}_1$ and additive *noise* $\mathbf{n}$ due to measurement uncertainties. Physical models of digital camera sensors, both for CCD and CMOS sensors, suggest that the readout noise $\mathbf{n}$ is a mixture of luminance independent (Gaussian, thermal) noise, and luminance dependent (Poisson or photon counting) noise [Healey and Kondepudy 1994; Tian et al. 2001; Marion 1997]. A usual simplification of this model, which we follow here, assumes the noise is image independent, white and Gaussian, with constant variance.

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

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

where $g(\cdot)$ is a monotone non-decreasing function that describes the non-linear sensor response. If the camera is working outside the saturation zone, in RAW images this response can be reasonably assumed to be linear (see Figure 2.14 in the Experimental section of Chapter 2). This boils down to a rescaling of the dynamics of u and therefore disappears with out loss of generality from the model. Hence, in what follows, the image formation model will be

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

3.2.2 Inverse problem statement in terms of digital sequences

Since in practice our data consist exclusively of discrete sequences (or digital images), the image formation model will be rewritten in terms of discrete sequences. This requires the introduction of additional notation, summarized in Table 3.1 (a more precise definition of each term is presented in Appendix 3.A). It would be cumbersome to verify systematically all regularity requirements on all functions and distributions needed in the proofs. Thus, all necessary results are given in a precise form in the appendices. They will be invoked in the proofs, and the reader is invited to check that their use was licit.

Suppose that the PSF h is s-band-limited, that is, $\text{supp}(\hat{h}) = [-s\pi, s\pi]^2$. Then, if sampled at a rate s, the Nyquist sampling theorem guarantees perfect reconstruction of h from its samples $\mathbf{h} = \mathbf{S}_1 H_{\frac{1}{s}} h$. We are actually interested in the case $s > 1$, usual for digital cameras. This means that the images obtained from (M) may be subject to aliasing.

Proposition 2 (discrete camera model). *Let* $u \in \mathcal{BL}_0^2$ *and* $h \in L^1 \cap \mathcal{BL}_0^2$*, band-limited in* $[-s\pi, s\pi]^2$*. Then*

$$\mathbf{S}_1(u * h) = \mathcal{S}_s(\bar{\mathbf{u}} * \mathbf{h}), \tag{3.1}$$

where we have called $\bar{\mathbf{u}} = \mathbf{S}_1 W_1 H_{\frac{1}{s}} u$ *and* $\mathbf{h} = \mathbf{S}_1 H_{\frac{1}{s}} h$.

u, v	Images defined on continuous domain $\mathbf{x} \in \mathbb{R}^2$
$\mathbf{u}, \mathbf{v}$	Digital images are sampled on a discrete grid $\mathbf{k} \in \mathbb{Z}^2$
$\mathbf{F}$	Fourier transform
$\hat{f}$	Fourier transform of a function f
$\mathbf{I}_1$	Shannon-Whittaker interpolator: $\mathbf{I}_1\mathbf{u}(\mathbf{x}) = \sum_{\mathbf{k}} \mathbf{u}(\mathbf{k})\text{sinc}(\mathbf{x} - \mathbf{k})$
$\mathbf{S}_1$	1-sampling operator: $\mathbf{u}(\mathbf{k}) = (\mathbf{S}_1 u)(\mathbf{k}) = u(\mathbf{k})$
W_w	Ideal low-pass filter cuts the spectrum of continuous signals to $[-w\pi, w\pi]^2$
$\mathcal{S}_s$	The s-to-1-resampling operator $\mathcal{S}_s = \mathbf{S}_1 H_s \mathbf{I}_1$
H_λ	Continuous homothecy: $H_\lambda u(x, y) = \lambda^2 u(\lambda x, \lambda y)$. ($\lambda < 1$ dilation)
$\mathbf{H}_\alpha$	Digital Nyquist homothecy operator of parameter α: $\mathbf{H}_\alpha \mathbf{u} := \mathbf{S}_1 W_1 H_\alpha \mathbf{I}_1 \mathbf{u}$
$\mathbf{C}[\mathbf{u}]$	Linear map associated to the convolution with a digital image $\mathbf{u}$
$\mathbf{L}^*$	Adjoint of a linear operator $\mathbf{L}$
$\mathbf{L}^+$	Pseudo-inverse $\mathbf{L}^+ := (\mathbf{L}^*\mathbf{L})^{-1}\mathbf{L}^*$ of a linear operator $\mathbf{L}$
L^1	Integrable functions on $\mathbb{R}^2$ ($L^1(\mathbb{R}^2)$)
L^2	Square integrable functions ($L^2(\mathbb{R}^2)$)
$\mathcal{BL}^2$	L^2 functions, band-limited in $[-\pi, \pi]^2$
$\mathcal{BL}^2_0$	L^2 functions with compactly supported Fourier transform

Table 3.1: Summary of the notation used in this chapter. A more precise definition of each term is presented in Appendix 3.A

Proof. We first derive the expression and then justify the application of each result. The set of used properties are detailed in the Appendices 3.A and 3.B.

$$\begin{aligned}
\mathbf{S}_1(u * h) &= \mathbf{S}_1 H_s H_{\frac{1}{s}}(u * h) \overset{(3.19)}{=} \mathbf{S}_1 H_s H_{\frac{1}{s}}(W_s u * h) \\
&\overset{(3.21)}{=} \mathbf{S}_1 H_s (H_{\frac{1}{s}} W_s u * H_{\frac{1}{s}} h) \overset{(3.20)}{=} \mathbf{S}_1 H_s (W_1 H_{\frac{1}{s}} u * H_{\frac{1}{s}} h) \\
&\overset{(3.17)}{=} \mathbf{S}_1 H_s \mathbf{I}_1 \mathbf{S}_1 (W_1 H_{\frac{1}{s}} u * H_{\frac{1}{s}} h) \overset{(3.22)}{=} \mathbf{S}_1 H_s \mathbf{I}_1 (\mathbf{S}_1 W_1 H_{\frac{1}{s}} u * \mathbf{S}_1 H_{\frac{1}{s}} h) \\
&\overset{\text{def}}{=} \mathbf{S}_1 H_s \mathbf{I}_1 (\bar{\mathbf{u}} * \mathbf{h}) \overset{\text{def}}{=} \mathcal{S}_s(\bar{\mathbf{u}} * \mathbf{h}).
\end{aligned}$$

First note that as $u \in \mathcal{BL}^2_0$ and $h \in L^1$ are band-limited in $[-s\pi, s\pi]^2$, we can apply (3.19) and (3.21) directly. As $W_1 u$ is in $\mathcal{BL}^2$ we can apply (3.20). The Nyquist theorem (3.17) is valid since $u \in L^2$ and $h \in L^1$, then $W_1 H_{\frac{1}{s}} u * H_{\frac{1}{s}} h$ belongs to $\mathcal{BL}^2$.

Both $W_1 H_{\frac{1}{s}} u$ and $H_{\frac{1}{s}} h$ are band-limited finite energy functions so we are free to apply (3.22). Since the sequence $(\bar{\mathbf{u}} * \mathbf{h})$ is the sampling of the band-limited L^2 function $W_1 H_{\frac{1}{s}} u * H_{\frac{1}{s}} h$, it belongs to ℓ^2 (Lemma 6). Finally, the interpolation $\mathbf{I}_1(\bar{\mathbf{u}} * \mathbf{h})$ is well defined. □

The previous proposition shows that the image formation model (M) can be written in terms of discrete sequences.

$$\begin{aligned}
\tilde{\mathbf{v}} &= \mathcal{S}_s(\bar{\mathbf{u}}_D * \mathbf{h}) + \mathbf{n} \\
&= \mathcal{S}_s \mathbf{C}[\bar{\mathbf{u}}_D]\mathbf{h} + \mathbf{n}. \qquad (3.2)
\end{aligned}$$

The digital image $\bar{\mathbf{u}}_D = \mathbf{S}_1 W_1 H_{\frac{1}{s}} u_D$ is a well-sampled version of the distorted image $u_D = u \circ D$. The value s is the resampling rate from the high resolution lattice $s\times$, where the PSF estimation will take place, to the $1\times$ sensor grid.

The numerical method will recover only a finite number of samples $\mathbf{h}$ of h. Strictly speaking h, being band-limited, cannot be compactly supported. Nonetheless, the error introduced by assuming that the support of h is bounded will prove negligible in comparison to the other sources of error: image noise, quantization, slight estimation errors of D, etc. Indeed, the retrieved solution $\mathbf{h}$ will prove to be experimentally independent from variations of its assumed support as long as it is large enough for errors to be negligible, and small enough for the operator to still be well conditioned.

When $\mathbf{n}$ is a zero-mean white discrete Gaussian noise, it follows from the previous formula that $\mathbf{h}_e = (\mathcal{S}_s\mathbf{C}[\bar{\mathbf{u}}_D])^+\tilde{\mathbf{v}}$ is an unbiased estimator of $\mathbf{h}$, as long as the linear operator $\mathcal{S}_s\mathbf{C}[\bar{\mathbf{u}}_D]$ is injective. It can be shown that the estimator variance is proportional to the Hilbert-Schmidt norm of $(\mathcal{S}_s\mathbf{C}[\bar{\mathbf{u}}_D])$ (for matrices, the Frobenius norm), and that it is nearly minimal when $\bar{\mathbf{u}}_D$ is a white noise realization (see Section 2.3).

3.3 PSF Estimation from an Unknown Pair of Scaled Images

Assume that we have perfect knowledge of the latent sharp image u that produced the blurry aliased observation $\tilde{\mathbf{v}}$. Under this non-blind assumption, solving for the PSF amounts to solving an inverse problem governed by the image formation model (M). Of course, this would require the use of a specially designed calibration pattern. We are now interested in investigating to what extent the use of such pattern could be circumvented. We will propose a method that allows us to accurately estimate the PSF from a pair of snapshots of the same scene, captured from different distances. In this method, the closest image will play a role similar to that of a calibration pattern in a classical non-blind approach.

In Chapter 2 we have shown that the highest accuracy in the PSF estimation is obtained by using a realization of a Bernoulli white noise as calibration pattern. However, many highly textured scenes do exist in nature which, while not being optimal, may still lead to a well-posed inverse problem. In what follows, we prove that from two far apart snapshots of this kind of scene, complete recovery of the camera PSF is theoretically possible based on the estimation of the blur between this pair.

3.3.1 Relative blur between two images: the inter-image kernel

Consider 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 and λ_2 denote the corresponding scale factors between the scene and each of the images. Then,

$$\begin{aligned}\tilde{\mathbf{v}}_i &= \mathbf{S}_1 H_{\lambda_i} u * h + \mathbf{n}_i \quad \text{for } i = 1, 2 \\ &= \mathbf{S}_1 v_i + \mathbf{n}_i \\ &= \mathbf{v}_i + \mathbf{n}_i,\end{aligned} \tag{3.3}$$

where $v_i := H_{\lambda_i} u * h$ and $\mathbf{v}_i := \mathbf{S}_1 v_i$. We will realistically assume that $h \in L^1 \cap \mathcal{BL}_0^2$ is nonnegative with $\|h\|_{L^1} = 1$, and $u \in \mathcal{BL}_0^2$ (details on the appropriateness of these assumptions are given in Appendix 3.A). Also, it will be assumed that the acquisition distances are such that $s\lambda_1 < \lambda_2$; the importance of this assumption will soon become clear.

Definition 2. *Let $v_1, v_2 \in \mathcal{BL}_0^2$ be two fronto-parallel continuous views of the same scene, acquired from different distances λ_1 and λ_2 respectively. We define an* ***inter-image kernel*** *between v_1 and v_2 as any kernel $k \in \mathcal{BL}_0^2$ satisfying*

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

The following lemma provides a characterization of the inter-image kernel.

Lemma 4. *Let $h \in L^1 \cap \mathcal{BL}_0^2$ be non-negative, band-limited with supp$(\hat{h}) \subset [-s\pi, s\pi]^2$ and $\hat{h}(0) = 1$. Let ρ be the largest positive number such that $|\hat{h}(\zeta)| > 0$ for every $\|\zeta\|_\infty < \rho\pi$ and assume that $\lambda_2 \rho > s\lambda_1$. Then there is an inter-image kernel $k \in \mathcal{BL}_0^2$ with support in $[-s\pi, s\pi]^2$ between (fronto-parallel views) v_1 and v_2 that satisfies*

$$H_\lambda h * k = h, \qquad \text{where } \lambda = \frac{\lambda_2}{\lambda_1}. \tag{3.4}$$

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 depends only on h and λ.

Proof. If k is an inter-image kernel between v_1 and v_2, according to Definition 2 it must satisfy

$$\mathbf{F}(H_\lambda v_1)(\zeta)\hat{k}(\zeta) = \hat{v}_2(\zeta).$$

Since $v_i := H_{\lambda_i} u * h$, the right-hand side of the previous equation is given by

$$\hat{v}_2(\zeta) = \hat{u}(\zeta/\lambda_2)\,\hat{h}(\zeta).$$

In the same way, for the left-hand side,

$$H_\lambda v_1 = H_\lambda(H_{\lambda_1} u * h) \overset{(3.21)}{=} H_{\lambda_2} u * H_\lambda h,$$

i.e., $\mathbf{F}(H_\lambda v_1)(\zeta)\hat{k}(\zeta) = \hat{u}(\zeta/\lambda_2)\,\hat{h}(\zeta/\lambda)$. Hence,

$$\hat{u}(\zeta/\lambda_2)\,\hat{h}(\zeta/\lambda)\,\hat{k}(\zeta) = \hat{u}(\zeta/\lambda_2)\,\hat{h}(\zeta). \tag{3.5}$$

It follows that a sufficient condition for k to be an inter-image kernel is $\hat{h}(\zeta/\lambda)\,\hat{k}(\zeta) = \hat{h}(\zeta)$. Since $h \in L^1$, $\hat{h}$ is continuous. It follows that ρ is necessarily positive, since $\hat{h}(0) = 1 > 0$. In addition, as $\lambda > \frac{s}{\rho}$ by hypothesis, $\mathbf{F}(H_\lambda h)(\zeta) = \hat{h}(\zeta/\lambda)$ does not vanish inside $[-s\pi, s\pi]^2$ and

$$\hat{k}(\zeta) = \frac{\hat{h}(\zeta)}{\hat{h}(\zeta/\lambda)} \tag{3.6}$$

is well defined all over its support, supp$(\hat{k}) \subset [-s\pi, s\pi]^2$. Finally, if $\hat{u}(\zeta/\lambda_2)$ does not vanish within the support of $\hat{h}$, from Eq. (3.5) k is unique. □

Remark 4. *In Lemma 4 it is assumed that the* PSF *h is the same for the two images. This has at least two practical implications. First, we assume that both images are taken in perfect focus through proper refocusing. The only camera parameter allowed to change is the focus (aperture and focal distance remain unchanged). Second, the common area between $\mathbf{v}_1$ and $\mathbf{v}_2$ covers an important part of $\mathbf{v}_1$, and consequently its* PSF *may exhibit some space variance that may degrade the estimation. Indeed, if v_1 is acquired through another* PSF *h' such that $\hat{h}'(\zeta) = \hat{h}(\zeta)$ in $[-s\frac{\pi}{\lambda}, s\frac{\pi}{\lambda}]^2$, then Equation (3.4) will still be valid. Thus, the real requirement is that the low frequencies of the* PSF *(i.e., frequencies in $[-s\frac{\pi}{\lambda}, s\frac{\pi}{\lambda}]^2$) do not change.*

3.3.2 Estimation of the inter-image kernel

The next goal is to estimate the inter-image kernel k. Since k is an s-band-limited function, we will work with its $s\times$ samples $\mathbf{k} = \mathbf{S}_1 H_{\frac{1}{s}} k$. We will show that under reasonable conditions, $\mathbf{k}$ can be recovered from the noisy aliased observations $\tilde{\mathbf{v}}_1$ and $\tilde{\mathbf{v}}_2$. Let us first build up some intuition on how to derive the proposed estimator. In what follows, $\mathring{\mathbf{v}}_1 = \mathbf{S}_1 W_1 H_{\frac{\lambda}{s}} v_1$ denotes a well sampled homothecy of parameter λ/s of v_1.

Proposition 3. *Under the assumptions of Lemma 4,*

$$\mathbf{v}_2 = (\mathcal{S}_s \mathbf{C}[\mathring{\mathbf{v}}_1])\mathbf{k}. \tag{3.7}$$

Proof. Being k an inter-image kernel between v_1 and v_2, it satisfies Eq. (3.4). Then,

$$\begin{aligned}\mathbf{v}_2 &= \mathbf{S}_1(v_2)\\ &= \mathbf{S}_1(H_\lambda v_1 * k).\end{aligned}$$

Since k is s-band-limited, it follows that

$$\mathbf{v}_2 \overset{(3.19)}{=} \mathbf{S}_1(W_s H_\lambda v_1 * W_s k). \tag{3.8}$$

Using the Nyquist-Shannon theorem for a band-limited signal and a set of properties detailed in Appendices 3.A and 3.B, yields

$$\begin{aligned}\mathbf{v}_2 &\overset{s\geq 0}{=} \mathbf{S}_1 H_s H_{\frac{1}{s}}(W_s H_\lambda v_1 * W_s k) \overset{(3.21)}{=} \mathbf{S}_1 H_s (H_{\frac{1}{s}} W_s H_\lambda v_1 * H_{\frac{1}{s}} W_s k)\\ &\overset{(3.20)}{=} \mathbf{S}_1 H_s (W_1 H_{\frac{\lambda}{s}} v_1 * W_1 H_{\frac{1}{s}} k) \overset{(3.17)}{=} \mathbf{S}_1 H_s \mathbf{I}_1 \mathbf{S}_1 (W_1 H_{\frac{\lambda}{s}} v_1 * W_1 H_{\frac{1}{s}} k)\\ &\overset{(3.22)}{=} \mathbf{S}_1 H_s \mathbf{I}_1 (\mathbf{S}_1 W_1 H_{\frac{\lambda}{s}} v_1 * \mathbf{S}_1 W_1 H_{\frac{1}{s}} k) \overset{\text{def}}{=} \mathbf{S}_1 H_s \mathbf{I}_1 (\mathring{\mathbf{v}}_1 * \mathbf{k})\\ &\overset{\text{def}}{=} \mathcal{S}_s(\mathring{\mathbf{v}}_1 * \mathbf{k}) \overset{\text{def}}{=} \mathcal{S}_s \mathbf{C}[\mathring{\mathbf{v}}_1]\mathbf{k}.\end{aligned}$$

□

Of course, in practice we do not have access to $\mathring{\mathbf{v}}_1$ or to $\mathbf{v}_2$, but only to their noisy, aliased versions $\tilde{\mathbf{v}}_1$ and $\tilde{\mathbf{v}}_2$. Thus $\mathbf{k}$ cannot be directly estimated from Eq. (3.7). However, a relationship between $\mathring{\mathbf{v}}_1$ and $\tilde{\mathbf{v}}_1$ can be established as follows:

$$\begin{aligned}\mathbf{H}_{\frac{\lambda}{s}} \tilde{\mathbf{v}}_1 &= \mathbf{H}_{\frac{\lambda}{s}}(\mathbf{v}_1 + \mathbf{n}_1) + \mathring{\mathbf{v}}_1 - \mathring{\mathbf{v}}_1\\ &= \mathring{\mathbf{v}}_1 + \underbrace{\mathbf{S}_1 W_1 H_{\frac{\lambda}{s}}(\mathbf{I}_1 \mathbf{v}_1 - v_1)}_{\mathbf{r}} + \mathbf{H}_{\frac{\lambda}{s}} \mathbf{n}_1,\end{aligned} \tag{3.9}$$

where the last equality results from the definition of the discrete homothecy operator. The term $\mathbf{r}$ is a consequence of aliasing when sampling v_1 and introduces an unknown bias in the estimation of $\mathbf{k}$. While this bias cannot be fully controlled, its impact can be mitigated. Indeed, since

$$\begin{aligned}\mathbf{r} &= \mathbf{S}_1 W_1 H_{\frac{\lambda}{s}}(\mathbf{I}_1 \mathbf{v}_1 - v_1)\\ &\overset{(3.20)}{=} \mathbf{S}_1 H_{\frac{\lambda}{s}} W_{\frac{s}{\lambda}}(\mathbf{I}_1 \mathbf{v}_1 - v_1),\end{aligned}$$

the aliasing term $\mathbf{r}$ will be non-zero only if there are aliasing components in the frequency interval $\left[-\frac{s}{\lambda}\pi, \frac{s}{\lambda}\pi\right]^2$. This allows us to choose $v_1 = H_{\lambda_1} u$ such that $\operatorname{supp}(\hat{v}_1) \subset \left[-2\pi + \frac{s}{\lambda}\pi, 2\pi - \frac{s}{\lambda}\pi\right]^2$ (see Figure 3.1). Thus, to minimize the impact of the aliasing term the images should be acquired from a pair of fronto-parallel locations as far as possible one from the other, since that amounts to increasing the value of λ.

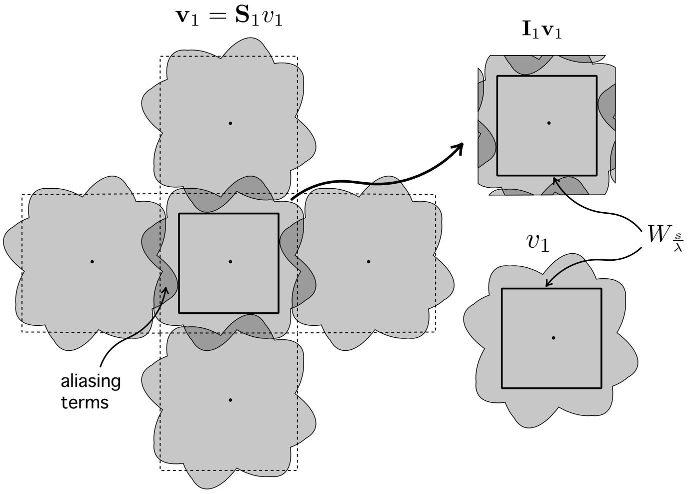


Figure 3.1: Neglecting the aliasing. The estimation will be affected by aliasing only if there are aliasing components in the interval $\left[-\frac{s}{\lambda}\pi, \frac{s}{\lambda}\pi\right]^2$. Hence, to avoid aliasing one can choose $v_1 = H_{\lambda_1} u$ such that $\operatorname{supp}(\hat{v}_1) \subset \left[-2\pi + \frac{s}{\lambda}\pi, 2\pi - \frac{s}{\lambda}\pi\right]^2$.

From now on, we assume that the snapshots are acquired following the previous considerations. Therefore, we can ignore the aliasing term in (3.9), which leads to

$$\mathbf{v}_2 = (\mathcal{S}_s \mathbf{C}[\mathring{\mathbf{v}}_1])\mathbf{k} = (\mathcal{S}_s \mathbf{C}[\mathbf{H}_{\frac{\lambda}{s}} \mathbf{v}_1 - \mathbf{H}_{\frac{\lambda}{s}} \mathbf{n}_1])\mathbf{k},$$

that is

$$(\mathcal{S}_s \mathbf{C}[\mathbf{H}_{\frac{\lambda}{s}} \tilde{\mathbf{v}}_1 - \mathbf{H}_{\frac{\lambda}{s}} \mathbf{n}_1])\mathbf{k} = \tilde{\mathbf{v}}_2 - \mathbf{n}_2.$$

One could be tempted to solve for $\mathbf{k}$ in the previous equation using a total least squares based approach:

$$\underset{\mathbf{k},\boldsymbol{\delta},\boldsymbol{\epsilon}}{\arg\min}\ \|\boldsymbol{\delta}\| + \kappa\|\boldsymbol{\epsilon}\| \ \text{ subject to } \ \mathcal{S}_s \mathbf{C}[\mathbf{H}_{\frac{\lambda}{s}} \tilde{\mathbf{v}}_1 + \boldsymbol{\delta}]\mathbf{k} = \tilde{\mathbf{v}}_2 + \boldsymbol{\epsilon}. \tag{TLS}$$

However, the particular structure of the operator $\mathcal{S}_s \mathbf{C}[\mathbf{H}_{\frac{\lambda}{s}} \tilde{\mathbf{v}}_1 + \boldsymbol{\delta}]$ makes this problem a difficult one. Instead we prefer to follow a simpler approach, which results from neglecting the noise term $\mathbf{H}_{\frac{\lambda}{s}} \mathbf{n}_1$. This yields to the least squares estimation problem

$$\underset{\mathbf{k},\boldsymbol{\epsilon}}{\arg\min}\ \|\boldsymbol{\epsilon}\| \ \text{ subject to } \ \mathcal{S}_s \mathbf{C}[\mathbf{H}_{\frac{\lambda}{s}} \tilde{\mathbf{v}}_1]\mathbf{k} = \tilde{\mathbf{v}}_2 + \boldsymbol{\epsilon}, \tag{LS}$$

whose solution is given by

$$\mathbf{k}_e = \left(\mathcal{S}_s \mathbf{C}[\mathbf{H}_{\frac{\lambda}{s}} \tilde{\mathbf{v}}_1]\right)^+ \tilde{\mathbf{v}}_2. \tag{3.10}$$

If the noise $\mathbf{n}_1$ is small compared to $\mathbf{v}_1$, this solution would be very close to the one that would be obtained from Problem (TLS). If, in addition, $\mathbf{n}_2$ is small compared to $\mathbf{v}_2$, both solutions would be close to the actual inter-image kernel $\mathbf{k} = \left(\mathcal{S}_s\mathbf{C}[\mathring{\mathbf{v}}_1]\right)^+ \mathbf{v}_2$. This follows directly from the continuity and injectivity assumptions on $\mathcal{S}_s\mathbf{C}[\mathring{\mathbf{v}}_1]$, as a consequence of Lemma 8. This being said, we will consider the estimator of the inter-image kernel in Eq. (3.10).

Remark 5. *If $\lambda < s$, the convolution between $\mathbf{k}$ and $\mathbf{H}_{\frac{\lambda}{s}}\tilde{\mathbf{v}}_1$ is not invertible so the operator $\mathcal{S}_s\mathbf{C}[\mathbf{H}_{\frac{\lambda}{s}}\tilde{\mathbf{v}}_1]$ will not be injective. This constraint on λ is necessary but not sufficient to make $\mathcal{S}_s\mathbf{C}[\mathbf{H}_{\frac{\lambda}{s}}\tilde{\mathbf{v}}_1]$ invertible. In addition, it is required that the spectrum of the image $\mathbf{H}_{\frac{\lambda}{s}}\tilde{\mathbf{v}}_1$ exhibits slow decay. Indeed, as shown in Section 2.3, the flatter the spectrum of the image scene is, the better conditioned is the inverse problem. For that reason, in order to obtain accurate estimates of $\mathbf{k}$, it is desirable that the chosen scene u exhibits white noise characteristics.*

3.3.3 From relative to absolute blur

Now that we have a method for estimating the inter-image kernel $\mathbf{k}$, we will concentrate on how to recover the camera PSF. Notice that h is related to k by $H_\lambda h * k = h$, and therefore its derivation is not straightforward. However, it holds that

$$h = \lim_{n\to\infty} H_{\lambda^{n-1}}k * H_{\lambda^{n-2}}k * \cdots * H_\lambda k * k, \tag{3.11}$$

as proved in the following proposition.

Proposition 4. *Let $h \in L^1 \cap \mathcal{BL}_0^2$ and $k \in \mathcal{BL}_0^2$ such that $\hat{k}(\zeta) = \frac{h(\zeta)}{h(\frac{\zeta}{\lambda})}$. Assume λ large enough to ensure that $\hat{h}(\zeta/\lambda)$ does not vanish in the support of $\hat{k}$. Then if $\lambda > 1$, we have*

$$\lim_{n\to\infty} H_{\lambda^{n-1}}k * H_{\lambda^{n-2}}k * \cdots * H_\lambda k * k = h,$$

where the limit is in $L^2 \cap C^0$.

Proof. Let us call $u_n = H_{\lambda^{n-1}}k * \cdots * H_\lambda k * k$. Then in the Fourier domain we have

$$\begin{aligned}\lim_{n\to\infty} \hat{u}_n(\zeta) &= \lim_{n\to\infty} \prod_{i=0}^{n-1} \hat{k}\left(\frac{\zeta}{\lambda^i}\right) \\ &= \lim_{n\to\infty} \frac{\hat{h}(\zeta)}{\hat{h}(\zeta/\lambda^n)}.\end{aligned}$$

Since $h \in L^1$, then $\hat{h} \in C^0$ and we have

$$\lim_{n\to\infty} \hat{h}(\zeta/\lambda^n) = \hat{h}(0) = 1$$

The convergence is uniform on a fixed compact set because $\hat{h}$ is continuous and compactly supported. This implies that the convergence holds in L^1 and L^2. Therefore

$$H_{\lambda^{n-1}}k * H_{\lambda^{n-2}}k * \cdots * H_\lambda k * k \xrightarrow{L^2\cap C^0} h.$$

□

This limit shows that it is possible to recover the camera PSF h from the inter-image kernel k. Recall that in practice we have access only to discrete sequences; therefore it is convenient to derive a discrete equivalent of the previous limit. Since $\mathbf{k}$ is s-band-limited,

$$\begin{aligned}\mathbf{S}_1 H_{\frac{1}{s}}(H_\lambda k * k) &\overset{(3.19)}{=} \mathbf{S}_1 H_{\frac{1}{s}}(W_s H_\lambda k * k) \overset{(3.21)}{=} \mathbf{S}_1(H_{\frac{1}{s}} W_s H_\lambda k * H_{\frac{1}{s}} k)\\ &\overset{(3.20)}{=} \mathbf{S}_1(W_1 H_{\frac{\lambda}{s}} k * H_{\frac{1}{s}} k) \overset{(3.1)}{=} \mathbf{S}_1 W_1 H_{\frac{\lambda}{s}} k * \mathbf{S}_1 H_{\frac{1}{s}} k\\ &\overset{\text{def}}{=} \mathbf{H}_\lambda \mathbf{k} * \mathbf{k}.\end{aligned}$$

Iteratively applying this result to Eq (3.11) yields

$$\mathbf{h} = \lim_{n\to\infty} \mathbf{H}_{\lambda^{n-1}}\mathbf{k} * \mathbf{H}_{\lambda^{n-2}}\mathbf{k} * \cdots * \mathbf{H}_\lambda \mathbf{k} * \mathbf{k}. \tag{3.12}$$

3.4 The Complete PSF Estimation Procedure

This section describes the algorithmic steps that lead to local subpixel PSF estimates. The complete chain is summarized in the block diagram of Figure 3.2. The next paragraphs present brief summaries for each block. A complete algorithmic description is given in Appendix A.2.

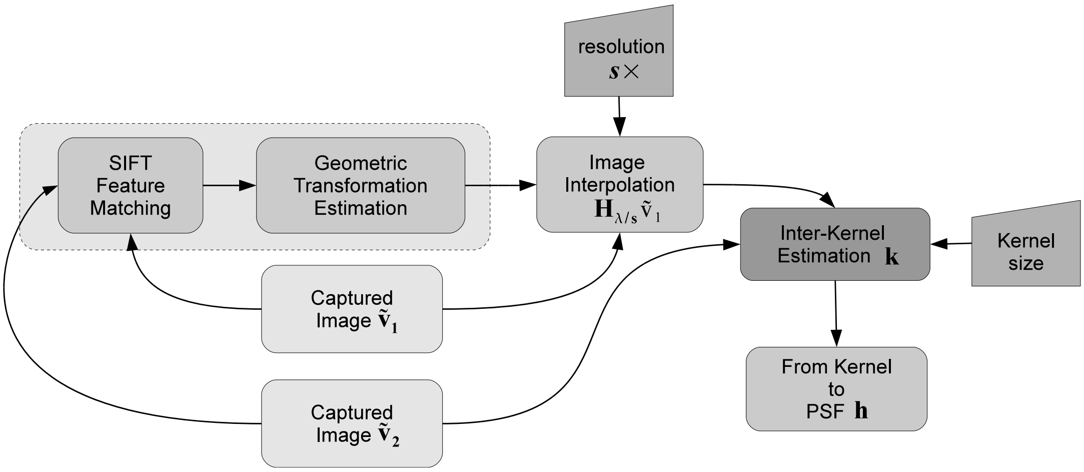


Figure 3.2: Algorithm description. Both captured images are aligned via SIFT feature matching followed by the estimation of a homography. The relative geometric transformation and gray-level corrections are applied to a low-pass (unaliased) version of the finest scale image $\tilde{\mathbf{v}}_1$. Then the interpolated image $\mathbf{H}_{\frac{\lambda}{s}}\tilde{\mathbf{v}}_1$ and image $\tilde{\mathbf{v}}_2$ are compared to obtain the inter-image kernel k, which is later iteratively updated to obtain the absolute camera PSF h.

Image alignment In order to estimate the geometric transformation between both images, they need to be precisely aligned. This alignment can be obtained by matching SIFT descriptors [Lowe 1999], which have the advantage of being scale invariant.

Geometric transform estimation The complete geometric transformation from one image to the other was approximated with a homography from the matched SIFT pairs. This permits the correction for deviations from the fronto-parallel assumption in the acquisition. Of course, if the distortion is significant the assumed inter-image kernel Eq. (3.4) will not be accurate. The relative scale $\lambda = (\lambda_x, \lambda_y)$ is taken directly from the estimated homography. The IPOL implementation by Moisan et al. [2012] was chosen because of the efficiency of the optimized random sampling algorithm (ORSA) rejection of false matches.

Gray level adjustment Both snapshots should be acquired with exactly the same camera configuration and constant scene illumination. This ensures that there is no contrast change between them.

Resampling and distortion correction of $\tilde{\mathbf{v}}_1$ The generation of the rescaled samples $\mathbf{H}_{\frac{\lambda}{s}}\tilde{\mathbf{v}}_1$ requires the interpolation of $\tilde{\mathbf{v}}_1$ at the desired scale λ/s. This is done by using the estimated geometric transformation with bicubic interpolation. Notice that since $\tilde{\mathbf{v}}_1$ is not very aliased, one can correctly interpolate it without introducing artifacts.

Numerical methods for inter-image kernel estimation Suppose that the image $\tilde{\mathbf{v}}_2$ has size $m \times n$. The goal is to estimate k at $s\times$ the resolution of $\tilde{\mathbf{v}}_2$ (camera sensor resolution). Also suppose that the estimated support of the inter-image kernel k is contained in an $r \times r$ patch. Then the matrix $\mathcal{S}_s\mathbf{C}[\mathbf{H}_{\frac{\lambda}{s}}\tilde{\mathbf{v}}_1]$ is of size $mn \times r^2$. A simple least squares procedure yields the inter-image kernel estimator:

$$\mathbf{k}_e = \arg\min_{\mathbf{k}} \left\| \mathcal{S}_s\mathbf{C}[\mathbf{H}_{\frac{\lambda}{s}}\tilde{\mathbf{v}}_1]\mathbf{k} - \tilde{\mathbf{v}}_2 \right\|^2 .$$

Transforming the kernel: from k to h Recovering the samples of the camera PSF $\mathbf{h}$ amounts to evaluate the limit in Eq (3.12). Directly working with the digital sequences requires some care in how the successive convolutions are computed. Since $\lambda > 1$, the application of $\mathbf{H}_\lambda$ would require a low-pass filter to avoid aliasing artifacts. To bypass this inconvenience one can restate the limit convolution as follows:

$$\mathbf{h} = \lim_{n\to\infty} \mathbf{H}_{\lambda^n}(\mathbf{k} * \mathbf{H}_{\frac{1}{\lambda}}\mathbf{k} * \cdots * \mathbf{H}_{\frac{1}{\lambda^{n-1}}}\mathbf{k} * \mathbf{H}_{\frac{1}{\lambda^n}}\mathbf{k}).$$

If implemented in this way, the successive discrete convolutions can be computed without any special care. To apply the discrete homothecy operator to $\mathbf{k}$, we need to resample $\mathbf{k}$ using the Shannon-Whittaker interpolator. Because of its slow decay, in order to reduce ringing and other windowing effects, we opted to use bicubic interpolation. We get $\mathbf{h}$ by an iterative procedure that converges after a few iterations since λ^n grows very fast. See Appendix A.2 for the algorithmic details.

In theory, as we already stated, the estimated PSF should be non-negative. In practice, small negative values may be observed, due to deviations from model assumptions and numerical artifacts. To correct for these deviations, we simply set all negative values to zero.

3.5 Experimental Results

Since there is no PSF ground truth available, the validation of the proposed method was carried out by simulations and by comparing the results with state-of-the-art methods Joshi et al. [2008]; LLC [2010] and with the proposed non-blind PSF estimation algorithm presented in Chapter 2. Comparison was made only to non-blind, pattern based methods, as the accuracy of blind methods is significantly lower. A complete algorithmic description, an online demo facility, and a reference source code can be found at the IPOL workshop by Delbracio et al. [2012a].

3.5.1 Simulations as a sanity check

A synthetic random image u was generated and re-interpolated $4\times$ in order to get the "continuous" sharp homothecy of the image u. Next both images were convolved with a PSF-like kernel (in this case a Gaussian isotropic kernel), and down-sampled to get the respective observed digital images at the camera resolution (i.e., $1\times$). The kernel was chosen so that the low resolution image presents aliasing artifacts. By generating the views of u in this way, there are no aliasing artifacts in the closest image. This experiment was done as a sanity check of the proposed method. A $4\times$ kernel was estimated from the observed image pair. The results are shown in Figures 3.3 and 3.4.

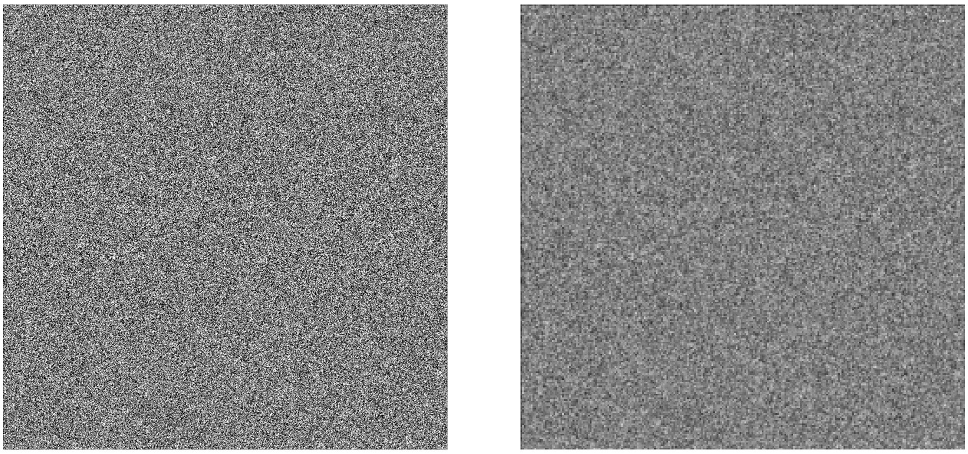
The procedure was tested for both automatic SIFT-based registration and the ideal (known) alignment. Although both estimates are significantly accurate, the automatic registration introduces a small misalignment, as shown in the difference images. See the caption of Figures 3.3 and 3.4 for details.

3.5.2 Real camera examples

The behavior of the proposed approach was tested for several different image pairs and for super-resolution estimations ranging from $1\times$ to $4\times$. The experiments were performed using a a Canon EOS 400D camera equipped with a Tamron AF 17-50mm F/2.8 XR Di-II lens. The focal length was fixed to 50.0 mm.

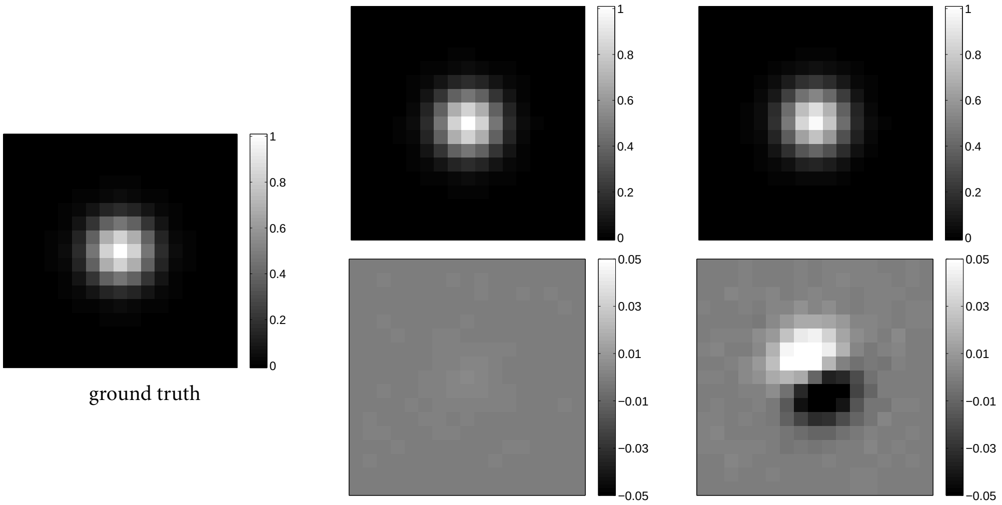
Two-scale versus non-blind pattern based method In Chapter 2 we presented a non-blind method that uses a realization of white noise as calibration pattern. It was proved that, until to now, this method is the one tat estimates the PSF with highest accuracy. Therefore, the PSF resulting from this method will be used here as ground truth. Figure 3.5 shows the $4\times$ PSF estimated by the proposed two-scale method from a pair of views of a wall shown in Figure 3.6. The estimation was conducted for one of the green channels (half of the green pixels of the Bayer matrix), with the camera aperture set to f/5.6. The estimated PSF is quite close to the one obtained by using our Bernoulli pattern based approach. In particular their sizes are similar, and their corresponding MTFS present zeros at the same locations.

Color filter array estimations Two pictures of another textured wall shown in Figure 3.7 were used to estimate the PSF of the four color Bayer channels (RAW camera output). This wall texture presents characteristics similar to those of white noise. The results for the $4\times$ PSF estimated at the image center are shown in Figure 3.7. Notice that the red channel PSF is wider than the green and the blue one, as expected from the physics of diffraction-limited optical systems, since the wavelengths associated to red light are larger than the rest. The differences between the dominant orientations of the red, green and blue PSFS spectra can be explained by the sensor shape and layout. In fact, each sensor active zone is usually L-shaped, and the red and blue sensors are rotated 90° w.r.t. the green ones (see, for example, Yadid-Pecht [2000]). These rotations are consistently observed in the PSFS and MTFS estimated with our two-scale method. This clearly illustrates the accuracy of the proposed approach.



closest image

farthest image (rescaled)



ground truth

ideal alignment

SIFT based alignment

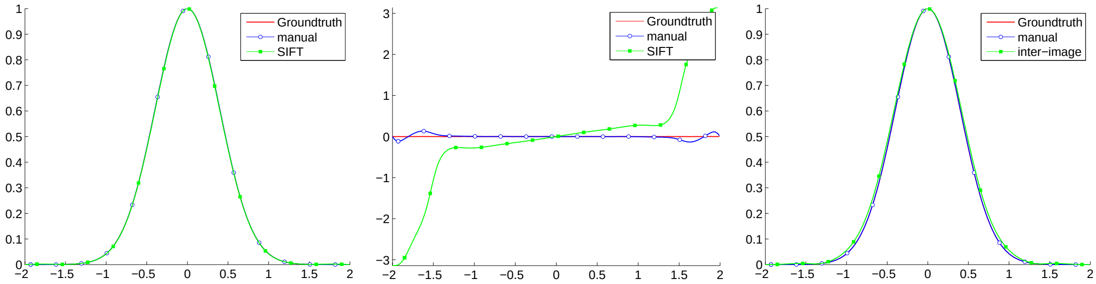
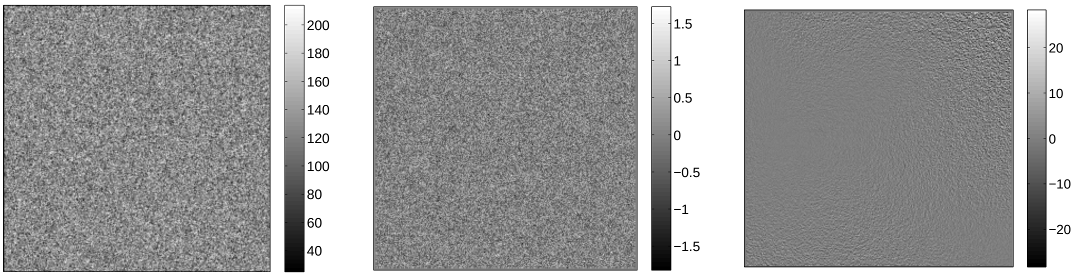


Figure 3.3: Synthetic example: $4\times$ PSF estimation for simulated data. Top row: the closest and farthest images. Middle row: the simulated PSF (ground truth) and the respective PSF estimations using the automatic SIFT points / homography alignment and the ideal alignment. Both estimations are accurate. However, as shown in the difference images the automatic registration introduces a small misalignment. This can also be seen in the phase and modulus of the PSF Fourier transform vertical profile, shown in the bottom row. Bottom row (right): comparison of the inter-image and PSF kernels. Since both input images are simulated at distances in a ratio of $\lambda = 4\times$, h is very close to k.



farthest image | image difference, ideal alignment | image difference, SIFT alignment

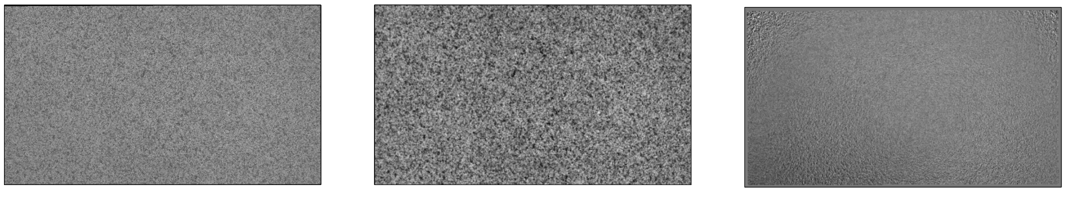
Figure 3.4: Synthetic example: $4\times$ PSF estimation for simulated data, residual image. From left to right: farthest image and residual images $\mathcal{S}_s(\mathbf{H}_{\frac{\lambda}{s}}\tilde{\mathbf{v}}_1 * \mathbf{k}) - \tilde{\mathbf{v}}_2$ with the estimated kernel from ideal and automatic alignment. The residual in the automatic alignment case is significantly larger than in the ideal alignment case. However, the difference in the PSFs seems to be negligible up to a subpixel translation as shown in Figure 3.3.

Different kinds of scenes The wall images in the previous experiments are well adapted for our two-scale PSF estimation method, since their spectra show slow decay. A priori one would think that images from pure white noise would yield better estimates, since this is what happens in our previous pattern based approach. But for our two-scale approach, this would be true if both snapshots could be precisely aligned, which is not the case in practice. Indeed, SIFT descriptors are not stable in the presence of aliasing. Hence, there is a trade-off between having accurate SIFT matches and textures with high frequency information. The texture shown in Figure 3.7 is an example of an appropriate trade-off.

Figure 3.8 shows two snapshots of a photograph in a magazine, with the corresponding $1\times$ to $4\times$ PSF estimations for the first green channel. The estimation was performed at the image center for the camera working at f/5.6 aperture. All the subpixel estimations are consistent: their MTFs exhibit good overlap in common regions. While these newspaper images produce accurate SIFT points, their spectra decay faster than those of the wall images. Consequently, the high frequencies in the PSF estimate are noisier. This can be readily seen by comparing both estimates at $4\times$ resolution.

What kind of textures should be used? It follows from the previous analysis that, in order to simultaneously produce good SIFT points and a sufficiently slow frequency decay, textures composed of elements with different sizes are to be preferred. Three-dimensional (3D) textures like those shown in Figure 3.9 can be problematic for this approach. Even though they respect the two previous conditions, their 3D nature produces disparities, and occlusions which change the image beyond a simple zoom. Likewise, non-Lambertian surfaces and dynamical scenes are not appropriated either.

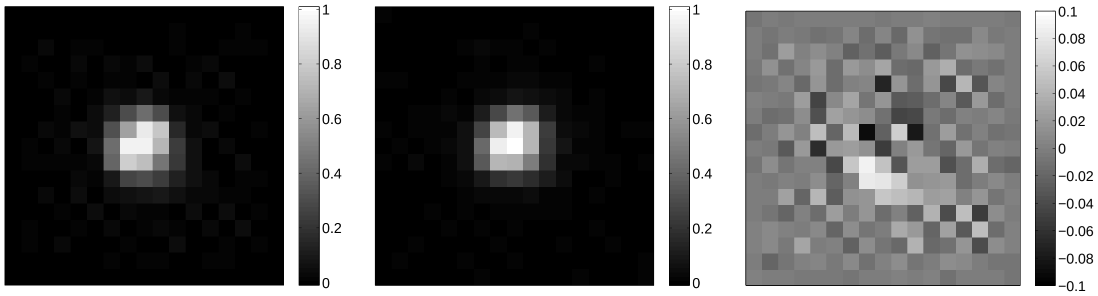
Comparison to other methods In this experiment we compare the performance of the two-scale method proposed here with three state-of-the-art non-blind methods: that of Joshi et al. [2008], Imatest commercial software LLC [2010], and our previous Bernoulli pattern method. All the estimates were computed at the image center, with aperture f/5.6. For the two-scale approach, we used the wall image pair shown in Figure 3.7. Joshi and coauthors and Imatest use two different kinds of slanted-edge calibration patterns. The algorithm by Joshi requires to set a regularization parameter; we show the results obtained for three different levels of regularization.



closest image (green v1 channel)

farthest image (green v1 channel)

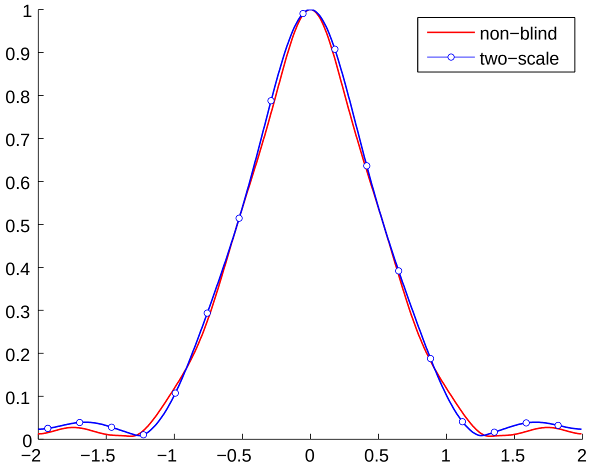
difference



Two scaled photographs

Bernoulli pattern based

difference



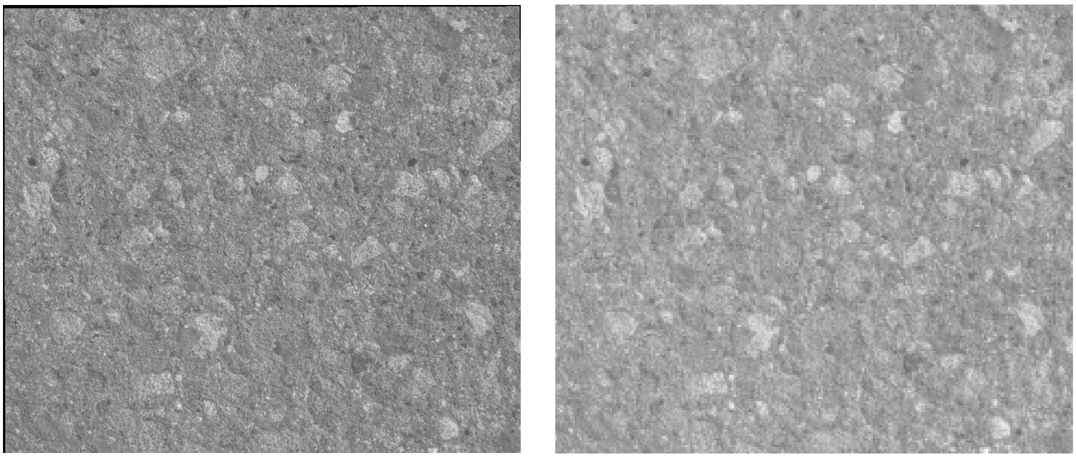
vertical profile MTF, (max difference < 0.04)

Figure 3.5: Wall image: two-scale versus Bernoulli white noise pattern based estimation. Estimation at $4\times$ PSF resolution for one of the green channels from the camera RAW output. Top row: two distant, parallel views of a textured wall. Middle row: the PSF estimated with the proposed algorithm and the one estimated using the Bernoulli pattern method. Bottom row: vertical profile of the MTF. Both estimations are close. In particular the associated airy disks have similar sizes, and the MTFs vanish in the same locations.

farthest snapshot

closest snapshot

Figure 3.6: Wall image: two-scale versus white noise pattern based estimation. Two distant, parallel views of a textured wall.



closest image

farthest image (close up)

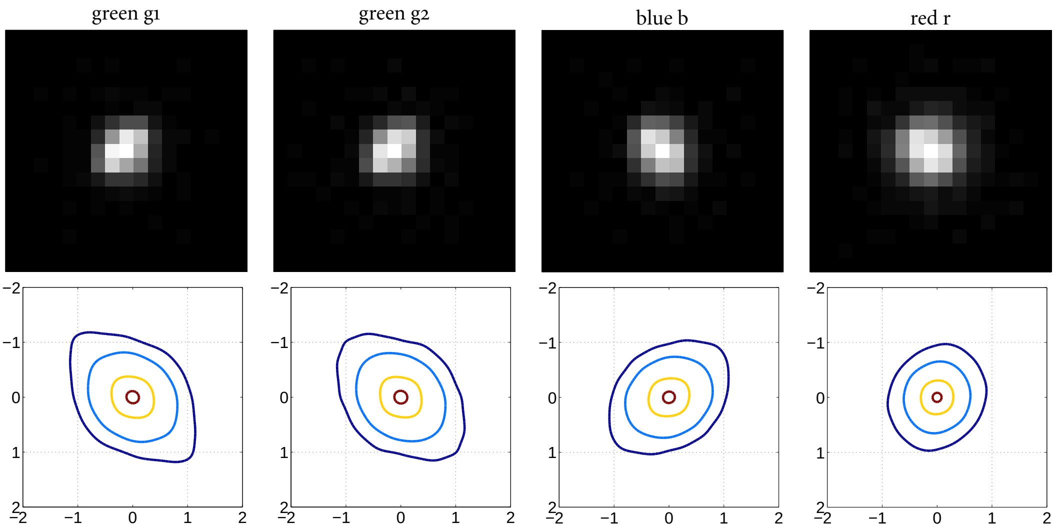
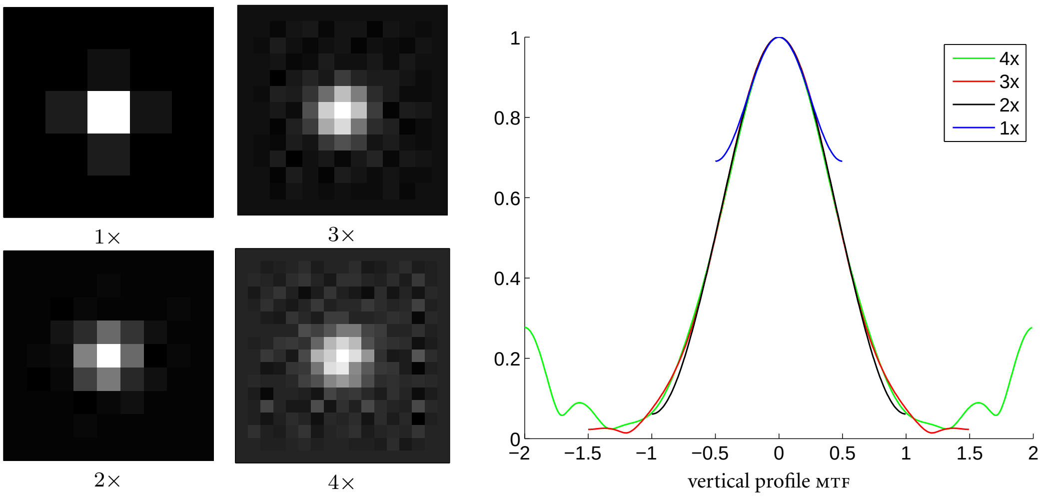


Figure 3.7: Different color channels. PSF estimation at $4\times$ resolution for the four Bayer channels (two greens, red and blue). Top row: two distant, parallel views of a concrete wall (close ups, the full images are shown in Figure 1.5). Middle row: the $4\times$ PSF estimated for the four channels. Bottom row: their corresponding Fourier spectrum modulus. The estimation was performed with images captured at aperture f/5.6. The red PSF is larger than the green and blue ones. This is consistent with the diffraction phenomenon: the red wavelengths are larger than the rest, thus the diffraction kernel is wider. Also note the differences between the shape of the red, blue and green PSF spectra (bottom row). Red and blue MTFs are rotated 90° with respect to the green ones. This symmetric behavior is consistent with the layout of L-shaped sensors Yadid-Pecht [2000].

closest photograph, rotated 90° farthest photograph, rotated 90°



1× 3× 2× 4×

Figure 3.8: Magazine image: $1\times$, $2\times$, $3\times$ and $4\times$ estimations for the first green channel from a pair of photographs of a newspaper image. The estimation was done at the image center for the camera working at an f/5.6 aperture. All the estimations are consistent: their MTFs show good overlap. The $4\times$ PSF estimation is noisier than the one produced from the wall images. The main reason is that the spectrum of the magazine image decays faster.

Figure 3.9: Examples of textures which are not adapted to the two-scale approach. Their 3D nature produces disparities and little changes in the angle-of-view would result in accuracy loss. Non-Lambertian surfaces and dynamical scenes are not appropriated either.

Figure 3.10 shows the MTF profiles of the obtained PSF estimates. The proposed two-scale method performs at least as well as the non-blind methods under comparison. The method of Joshi and colleagues shows similar performance for a carefully, manually chosen regularization parameter. See caption for details.

3.6 Conclusion

In this chapter we presented an algorithm for the subpixel estimation of the point spread function (PSF) 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 estimation method is regularization-free. In that sense, the technique is closely related to the non-blind estimation method presented in Chapter 2, which uses a random noise pattern. The main difference is that non-blind methods can directly estimate the PSF using the perfect knowledge of the pattern. In the proposed two-scale method the question is far more intricate because only the blur between the acquisitions can be estimated. Thus a mathematical analysis and new algorithms have been introduced proving how the PSF can be recovered from the inter-image kernel.

To reach high accuracy, images of textured scenes with sufficient flat spectra are preferred. It was experimentally verified that many textures found in nature are well adapted to these requirements. A comparison of the resulting PSF estimates with other subpixel PSF estimation methods shows that the proposed algorithm reaches accuracy levels similar to those of state-of-the-art methods, with the advantage of not requiring any special acquisition setup or calibration pattern and thus being much more practical.

3.A Mathematical Framework and Physical Modeling

Functional spaces and other notation

- $\mathbb{R}^2$ is the set of pairs of real numbers $\mathbf{x} = (x_1, x_2)$, and $\mathbb{Z}^2$ the set of pairs of integers $\mathbf{k} = (k_1, k_2)$. $L^1(\mathbb{R}^2)$ is the set of integrable functions on $\mathbb{R}^2$, $L^2(\mathbb{R}^2)$ is the set of square

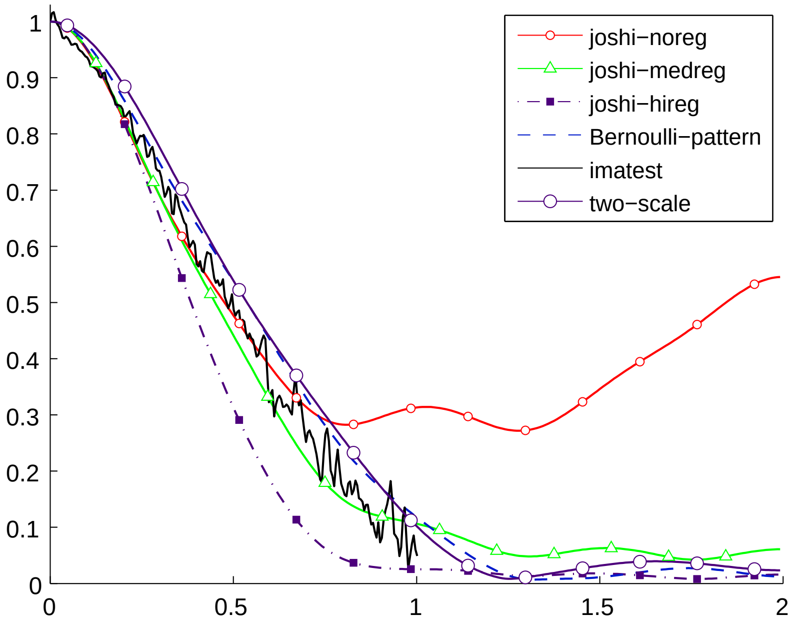


Figure 3.10: Comparison of PSF/MTF estimation methods. Our implementation of the PSF estimation algorithm by Joshi et al. [2008], Imatest commercial software LLC [2010], our previous Bernoulli pattern method, and the two-scale method proposed in this work (applied to the images of the wall shown in Fig. 3.7). On the low frequencies all algorithms produced very similar estimates, while on the higher frequencies the Joshi et al. estimation depends strongly on the regularization level. Although much effort was made to get a noise-free MTF estimation from the Imatest software, the final estimation is quite noisy. The Imatest estimation is done from a slanted-edge image and only gives an estimation for the MTF at the slanted-edge orthogonal direction. The proposed two-scale algorithm is the one presenting an estimation closest to the non-blind estimation presented in Chapter 2, considered as ground truth by virtue of its high accuracy.

integrable functions, $C^0_b(\mathbb{R}^2)$ is the set of continuous bounded functions, $C^\infty(\mathbb{R}^2)$ is the set of infinitely differentiable functions, $\mathcal{S}(\mathbb{R}^2)$ is the Schwartz class of C^∞ functions whose derivatives of all orders have fast decay, $\mathcal{S}'(\mathbb{R}^2)$ is its dual, the space of tempered distributions, $\mathcal{E}'$ the subset of $\mathcal{S}'(\mathbb{R}^2)$ of compactly supported distributions. We shall use the properties of the convolution $L^1 * L^2 \subset L^2$, $L^1 * L^1 \subset L^1$, $L^2 * L^2 \subset C^0$, $\mathcal{E}' * \mathcal{S}' \subset \mathcal{S}'$.

- We denote by $\mathcal{BL}^2(\mathbb{R}^2)$ (or $\mathcal{BL}^2$ for short) the set of L^2 functions that are band-limited inside $[-\pi, \pi]^2$. More generally, $\mathcal{BL}^2_0$ denotes the space of L^2 functions with compactly supported Fourier transform.

The following conventions and notations will be used in what follows:

- $\mathbf{F}$ is the *Fourier transform operator* defined on $\mathcal{S}'$; $\mathbf{F}(f)(\zeta) = \hat{f}(\zeta) = \int e^{-i\mathbf{x}\cdot\zeta} f(\mathbf{x}) d\mathbf{x}$ defines it for a function $f \in L^1(\mathbb{R}^2)$ in a point $\zeta = (\zeta_1, \zeta_2)$. This formula is still valid for functions belonging to $L^p(\mathbb{R}^2)$ with $1 < p \leq 2$ (see, e.g., the reference books by Stein and Weiss [1971]; Bony [2001]).

- *Continuous images* are defined for $\mathbf{x} \in \mathbb{R}^2$, whereas *digital images* are sampled on a discrete grid $\mathbf{k} \in \mathbb{Z}^2$. realistic assumption is to consider them non-negative.

- $\mathbf{S}_1 : C^0_b \to \ell^\infty(\mathbb{Z}^2)$ is the 1-*sampling operator* such that $\mathbf{u}(\mathbf{k}) = (\mathbf{S}_1 u)(\mathbf{k})$. From the distribution viewpoint $\mathbf{S}_1$ is the product by a Dirac comb $\Pi_s := \sum_{\mathbf{k}} \delta_{s\mathbf{k}}$ with $s = 1$, namely $\mathbf{S}_1 u = \Pi_1 . u$ where u must be a continuous function. Both representations of the sampling operator will be identified, and it will be clear from the context which representation is intended.

- A *digital image* $\mathbf{u}$ will be represented either as a sequence $(u(\mathbf{k}))_{\mathbf{k}}$ in $\ell^\infty(\mathbb{Z}^2)$ or as the corresponding Dirac comb $\mathbf{u} := \sum_{\mathbf{k}\in\mathbb{Z}^2} \mathbf{u}(\mathbf{k})\delta_{\mathbf{k}}$.

- The operator $\mathbf{I}_1 : \ell^2(\mathbb{Z}^2) \to \mathcal{BL}^2(\mathbb{R}^2)$ denotes the *Shannon-Whittaker interpolator*, defined by $\mathbf{I}_1\mathbf{u}(\mathbf{x}) = \sum_{\mathbf{k}\in\mathbb{Z}^2} \mathbf{u}(\mathbf{k})\text{sinc}(\mathbf{x} - \mathbf{k})$, where $\text{sinc}(\mathbf{x}) = \frac{\sin(\pi x)}{\pi x}\frac{\sin(\pi y)}{\pi y}$. We therefore have $\mathbf{I}_1\mathbf{u} = \mathbf{F}^{-1}(\sum_{\mathbf{k}} \mathbf{u}(\mathbf{k}) e^{-i\mathbf{k}\cdot\xi} \mathbf{1}_{[-\pi,\pi]^2})$. When $\mathbf{u} \in \ell^2$, $\mathbf{F}(\mathbf{I}_1\mathbf{u})$ belongs to L^2 and is compactly supported. Thus $\mathbf{I}_1\mathbf{u} \in \mathcal{BL}^2$, and we have $\mathbf{S}_1\mathbf{I}_1 = Id$.

- The filter $W_w u = \mathbf{F}^{-1}(\hat{u} \cdot \mathbf{1}_{[-w\pi, w\pi]^2})$ is an *ideal low-pass filter* that cuts the spectrum of u to $[-w\pi, w\pi]^2$. It is defined if $\hat{u}$ is a function. Note that if $u \in L^1 \cup L^2$, then $W_1 u$ is in $\mathcal{BL}^2$.

- $H_\lambda u(\mathbf{x}) = \lambda^2 u(\lambda \mathbf{x})$ is the continuous homothecy (i.e., $\lambda > 1$ is a contraction); the rationale for its normalization is to preserve the image mean (its zero-frequency coefficient). In the Fourier domain $\mathbf{F}(H_{\frac{\lambda}{u}})(\zeta) = \hat{u}(\frac{\zeta}{\lambda})$, so if u is α-band-limited, then $H_{\frac{1}{\alpha}} u$ is band-limited.

- $\mathcal{S}_s : \ell^2(\mathbb{Z}^2) \to \ell^2(\mathbb{Z}^2)$ denotes the s-to-1-resampling operator $\mathcal{S}_s = \mathbf{S}_1 H_s \mathbf{I}_1$ (i.e., $s > 1$ is a subsampling by s).

- $\mathbf{C}[\mathbf{u}] : \ell^2(\mathbb{Z}^2) \to \ell^2(\mathbb{Z}^2)$ denotes the linear map associated to the convolution with a digital image $\mathbf{u}$. The convolved sequence belongs to $\ell^2(\mathbb{Z}^2)$ which in general is satisfied if $\mathbf{u} \in \ell^1(\mathbb{Z}^2)$.

- The digital Nyquist homothecy operator $\mathbf{H}_\alpha : \ell^2(\mathbb{Z}^2) \to \ell^2(\mathbb{Z}^2)$ is defined by $\mathbf{H}_\alpha \mathbf{u} := \mathbf{S}_1 W_1 H_\alpha \mathbf{I}_1 \mathbf{u}$. It is a digital contraction if $\alpha > 1$.

- Let $\mathbf{L}$ be a bounded linear operator over a Hilbert space. $\mathbf{L}^*$ is its adjoint and $\mathbf{L}^+$ (if it exists) is its pseudo-inverse, i.e., the minimum-norm solution of $(\mathbf{L}^*\mathbf{L})\mathbf{L}^+ := \mathbf{L}^*$.

Physical and mathematical modeling of continuous images

Continuous images will be assumed to be functions in $\mathcal{BL}_0^2(\mathbb{R}^2)$. This choice is consistent, since these functions are continuous (actually C^∞) and the sampling is well defined. Moreover, as suggested by Morel and Ladjal [1998] and later by Almansa et al. [2004, Appendix A] this choice is sufficiently general to model the continuous landscape observed by a camera just before sampling takes place at the sensors.

In fact, even if the raw physical image before blur and sampling is, realistically, a positive Radon measure O (due to the photon-counting nature of sensitive digital systems) with compact support (imposed by the finite number of photons), it will still be blurred by the camera PSF h which will be regular enough for $h * O$ to be in $\mathcal{BL}_0^2$.

How regular can it realistically be assumed to be? The kernel h originates in several physical phenomena from diffraction, anti-aliasing filtering and sensor integration. Each one of these phenomena, and their combination as well, lead to model h as a nonnegative function with finite mass $\int h = 1$ (normalized to 1). In addition the diffraction part ensures that $\hat{h}$ is compactly supported. From this one deduces that $h \in \mathcal{BL}_0^2 \cap L^1$.

We now turn to the problem of simplifying O to a more manageable function u, which is indistinguishable from O after convolution with the PSF h. Let $B = \operatorname{supp}(\hat{h})$ be the (compact) spectral support of the PSF h. Hence h can be idempotently written as $h = h * h_0$, where $h_0 \in \mathcal{S}'$ has a compactly supported spectrum satisfying $\hat{h}_0(\eta) = 1$ for $\eta \in B$. The function $\hat{h}_0$ can easily constructed by an explicit formula as a C^∞ and compactly supported function satisfying $\hat{h}_0(\eta) = 1$ on B. Then its inverse Fourier transform has all required properties.

So we have

$$v = h * O = h * u, \quad \text{where } u = h_0 * O.$$

Consequently, the observed landscape can be assumed without loss of generality to be $u = h_0 * O$ instead of O. Being the convolution of a compactly supported positive Radon measure $O \in \mathcal{E}'$ with $h_0 \in \mathcal{BL}_0^2 \cap L^1$, u also belongs to $\mathcal{BL}_0^2$, and its convolution with $h \in \mathcal{BL}_0^2 \cap L^1$ is the observed image $v \in \mathcal{BL}_0^2$.

Standard results from Fourier analysis

The following two main results from standard Fourier analysis and distribution theory are stated without proof. The reader is referred to [Stein and Weiss 1971; Hörmander 1983] for the proofs in the particular setting chosen here.

Proposition 5 (convolution through Fourier transform). *The relation*

$$\mathbf{F}(f * g) = \mathbf{F}(f) \cdot \mathbf{F}(g) \tag{3.13}$$

is valid in either of these cases

1. *$g \in L^1(\mathbb{R}^2)$ and $f \in L^p(\mathbb{R}^2)$ for $1 \leq p \leq 2$. Then $f * g$ belongs to $L^p(\mathbb{R}^2)$ (see [Stein and Weiss 1971, Theorem 2.6]).*
2. *$g \in \mathcal{E}'$ and $f \in \mathcal{S}'$. Then $f * g$ belongs to $\mathcal{S}'$ (see [Hörmander 1983, Theorem 7.1.15]).*

Applying the Fourier transform on both sides of (3.13) and recalling that the squared Fourier transform operator $\mathbf{F}^2(u) = (2\pi)^2[\mathbf{x} \mapsto u(-\mathbf{x})]$ is almost the identity (except for flipping and a constant factor), we obtain the following:

Corollary 1 (product through Fourier transform). *The relation*

$$\begin{aligned} \mathbf{F}(f \cdot g) &= \frac{1}{(2\pi)^2}\mathbf{F}(f) * \mathbf{F}(g) \\ \mathbf{F}^{-1}(f \cdot g) &= \mathbf{F}^{-1}(f) * \mathbf{F}^{-1}(g) \end{aligned} \tag{3.14}$$

holds when $\hat{g} \in \mathcal{E}'$ and $f \in \mathcal{S}'$. Then $f \cdot g$ belongs to $\mathcal{S}'$.

Proposition 6 (Poisson formula in $\mathbb{R}^2$ for tempered distributions [Hörmander 1983]).

$$\hat{\Pi}_1 = (2\pi)^2 \Pi_{2\pi}. \tag{3.15}$$

Lemma 5. *If $\hat{u} \in \mathcal{E}'$, then*

$$\mathbf{F}(\Pi_1 \cdot u) = \Pi_{2\pi} * \hat{u}. \tag{3.16}$$

Proof. We can apply the first form of Corollary 1 where $f = \Pi_1 \in \mathcal{S}'$ and $\hat{g} = \hat{u} \in \mathcal{E}'$ to obtain

$$\mathbf{F}(\Pi_1 \cdot u) = (2\pi)^{-2}\hat{\Pi}_1 * \hat{u} = \Pi_{2\pi} * \hat{u}$$

where the last equality is deduced from the Poisson formula (3.15). □

The Shannon-Whittaker sampling theorem is then a direct consequence of the two previous results.

Proposition 7 (Nyquist-Shannon theorem). *If $u \in \mathcal{BL}^2(\mathbb{R}^2)$, then*

$$u = \mathbf{I}_1\mathbf{S}_1 u. \tag{3.17}$$

Proof. We can apply Lemma 5. Multiplying both sides of Equation (3.16) by $\mathbf{F}(\mathrm{sinc}) = \mathbb{1}_{[B]}$ we obtain

$$\begin{aligned} \mathbf{F}(\mathrm{sinc}) \cdot \mathbf{F}[\mathbf{S}_1 u] &= \mathbf{F}(\mathrm{sinc}) \cdot [\Pi_{2\pi} * \hat{u}] \\ &= \sum_{\mathbf{k} \in \mathbb{Z}^2} \hat{u}(\cdot + 2\pi\mathbf{k})\mathbb{1}_{[B]} \\ &= \hat{u}, \end{aligned}$$

where in the right-hand side the only non-null term is for $\mathbf{k} = 0$ because u is band-limited in $B = [-\pi, \pi]^2$ and $\mathbf{F}(\mathrm{sinc}) = \mathbb{1}_{[B]}$. Finally, using the second form of Corollary 1, we obtain

$$\mathrm{sinc} * (\mathbf{S}_1 u) = u,$$

and the left term is by definition $\mathbf{I}_1\mathbf{S}_1 u$. □

Corollary 2. *If $u \in L^2$ is s-band-limited, then*

$$u = H_s \mathbf{I}_1 \mathbf{S}_1 H_{\frac{1}{s}} u. \tag{3.18}$$

3.B Proof of Auxiliary Results

Common hypotheses According to the discussion in Appendix 3.A, and in order to justify all the lemmas and propositions we will require that

- $h \in \mathcal{BL}_0^2 \cap L^1(\mathbb{R}^2)$, non-negative $\hat{h}(0) = 1$;
- $u \in \mathcal{BL}_0^2$.

This ensures that the convolution $u * h = v$ is well defined with $u \in \mathcal{BL}_0^2$. For the uniqueness of the inter-image kernel we shall additionally assume that $\hat{u}$ does not vanish inside $[-\frac{s}{\lambda_2}\pi, \frac{s}{\lambda_2}\pi]^2$.

We now prove several properties that are used throughout this chapter.

Lemma 6. *If $u \in \mathcal{BL}_0^2$, then $\mathbf{S}_1 u \in \ell^2(\mathbb{Z}^2)$.*

Proof. As u is in $\mathcal{BL}_0^2$ there exists $s > 0$ such that $\hat{u} \subset [-s\pi, s\pi]^2$. Furthermore, since $\hat{u} \in \mathcal{E}'$ applying (3.16) we have $\mathbf{F}(\mathbf{S}_1 u) = (2\pi)^2 \Pi_{2\pi} * \hat{u}$. Since u belongs to L^2, then $\hat{u}$ is again in L^2. Thus, $\Pi_{2\pi} * \hat{u}$ is the 2π-periodic version of a L^2 function in $[-s\pi, s\pi]^2$. Consequently the inverse Fourier transform of $\Pi_{2\pi} * \hat{u}$ is a Dirac comb whose coefficients are the Fourier series coefficients of $\hat{u}$. Thus the coefficients of $\mathbf{S}_1 u$ form an ℓ^2 sequence. □

Proposition 8. *Let $h \in L^1(\mathbb{R}^2)$ and $u, v \in L^1 \cup L^2(\mathbb{R}^2)$. The following equalities hold:*

$$W_1(h * v) = W_1 h * v = h * W_1 v, \tag{3.19}$$

$$W_1 H_\lambda v = H_\lambda W_{\frac{1}{\lambda}} v, \tag{3.20}$$

$$H_\alpha(u * v) = H_\alpha u * H_\alpha v. \tag{3.21}$$

Proof. This is the proof of (3.19). In the Fourier domain,

$$\mathbf{F}(W_1(h * v)) \overset{\text{def}}{=} \mathbf{F}(h * v) \cdot \mathbf{1}_{[-\pi,\pi]^2} \overset{(3.13)}{=} \mathbf{F}(h) \cdot \mathbf{F}(v) \cdot \mathbf{1}_{[-\pi,\pi]^2}.$$

Thus,

$$\mathbf{F}(h) \cdot \mathbf{F}(v) \cdot \mathbf{1}_{[-\pi,\pi]^2} = \mathbf{F}(h) \cdot \mathbf{1}_{[-\pi,\pi]^2} \cdot \mathbf{F}(v) \cdot \mathbf{1}_{[-\pi,\pi]^2},$$

and all results are deduced from this last statement. □

Proof. This is the proof of (3.20). Since

$$\mathbf{F}(H_\lambda v) = \lambda^2 \mathbf{F}(v(\lambda \cdot)) = \lambda^2 \frac{1}{\lambda^2} \hat{v}(\frac{\cdot}{\lambda}) = \lambda^2 H_{\frac{1}{\lambda}} \hat{v},$$

we have

$$\mathbf{F}(W_1 H_\lambda v) \overset{(3.13)}{=} \mathbf{F}(H_\lambda v) \cdot \mathbf{1}_{[-\pi,\pi]^2} = \lambda^2 H_{\frac{1}{\lambda}} \hat{v} \cdot \mathbf{1}_{[-\pi,\pi]^2}.$$

On the other hand,

$$\begin{aligned} \mathbf{F}(H_\lambda W_{\frac{1}{\lambda}} u) &= \lambda^2 H_{\frac{1}{\lambda}} \mathbf{F}(W_{\frac{1}{\lambda}} v) \\ &\overset{(3.13)}{=} \lambda^2 H_{\frac{1}{\lambda}} (\hat{u} \cdot \mathbf{1}_{[-\frac{\pi}{\lambda}, \frac{\pi}{\lambda}]}) \\ &= \lambda^2 (H_{\frac{1}{\lambda}} \hat{v}) \cdot \mathbf{1}_{[-\pi,\pi]^2}. \end{aligned}$$

□

Proof. This is the proof of (3.21). The proof is a mere change of variables:

$$\begin{aligned} H_\alpha(u * v)(\mathbf{x}) &= \alpha^2 \int u(\mathbf{s})v(\alpha\mathbf{x} - \mathbf{s})\mathrm{d}\mathbf{s} \\ &= \alpha^4 \int u(\alpha\mathbf{s})v(\alpha\mathbf{x} - \alpha\mathbf{s})\mathrm{d}\mathbf{s} \\ &= (H_\alpha u * H_\alpha v)(\mathbf{x}). \end{aligned}$$

□

Lemma 7. *Let* $u, v \in \mathcal{BL}_0^2(\mathbb{R}^2)$. *If either* u *or* v *is band-limited, then*

$$\mathbf{S}_1(u * v) = \mathbf{S}_1\bar{u} * \mathbf{S}_1\bar{v}, \tag{3.22}$$

where we have called $\bar{u} = W_1 u$ *and* $\bar{v} = W_1 v$ *.*

Proof. We will prove this statement in the tempered distribution sense. We will consider $\mathbf{S}_1 u = \Pi_1 \cdot u = \sum_{\mathbf{k}} \delta_{\mathbf{k}} \cdot u$ as a Dirac comb. The application of $\mathbf{S}_1$ to $\bar{u}$, $\bar{v}$ and $u * v$ is well defined as all functions are in $\mathcal{BL}^2(\mathbb{R}^2)$ and by consequence they are in C^∞. Recall that if $u \in \mathcal{D}'$ and f is C^∞ then $f \cdot u \in \mathcal{D}'$ thus in this framework we need a function to be in C^∞ to be sampled.

From Lemma 6 we know that the sequence of coefficients from $\mathbf{S}_1\bar{u}$ and $\mathbf{S}_1\bar{v}$ are in $\ell^2(\mathbb{Z}^2)$. Thus $(\mathbf{S}_1\bar{u}) * (\mathbf{S}_1\bar{v})$ is a bounded sequence and therefore every term is well defined.

Finally $\mathbf{F}(\mathbf{S}_1(u * v)) = \Pi_{2\pi} * (\hat{u}.\hat{v}) = (\Pi_{2\pi} * \hat{u}) \cdot (\Pi_{2\pi} * \hat{v})$ is true because all considered functions happen to be 2π-periodizations of compactly supported functions in $(-\pi, \pi)^2$, namely $\hat{u}$, $\hat{v}$, and their product. □

Lemma 8 (stability of the inter image kernel estimation)**.** *Let* $\mathbf{A}$ *be an injective bounded linear operator (*IBLO*) defined on a Banach space* X *and* $\Delta\mathbf{A}$ *a perturbation of* $\mathbf{A}$ *such that* $\mathbf{A} + \Delta\mathbf{A}$ *is also* IBLO *and* $\|\mathbf{A}\|\|\Delta\mathbf{A}\| < 1$. *Let* $\mathbf{b} \in X$ *and* $\Delta\mathbf{b}$ *be a perturbation of* $\mathbf{b}$. *Then, the solution of* $\mathbf{x} = \mathbf{A}^+\mathbf{b}$ *and* $\mathbf{x}^* = (\mathbf{A} + \Delta\mathbf{A})^+(\mathbf{b} + \delta\mathbf{b})$ *satisfy:*

$$\frac{\|\mathbf{x}^* - \mathbf{x}\|}{\|\mathbf{x}\|} \leq \frac{\operatorname{cond}(\mathbf{A})}{1 - \|\mathbf{A}^+\Delta\mathbf{A}\|}\left(\frac{\|\delta\mathbf{b}\|}{\|\mathbf{b}\|} + \frac{\|\Delta\mathbf{A}\|}{\|\mathbf{A}\|}\right), \tag{3.23}$$

where $\operatorname{cond}(\mathbf{A}) = \|\mathbf{A}\|\|\mathbf{A}^+\|$.

Proof. First note that as $\mathbf{A}$ is full rank the pseudo-inverse is the left inverse of $\mathbf{A}$, namely $\mathbf{A}^+\mathbf{A} = \mathbf{I}$. Since $\|\mathbf{A}\|\|\Delta\mathbf{A}\| < 1$ we have that $(\mathbf{A} + \Delta\mathbf{A})^+ = (\mathbf{I} + \mathbf{A}^+\Delta\mathbf{A})^{-1}\mathbf{A}^+$ and we also have

$$\|(\mathbf{I} + \mathbf{A}^+\Delta\mathbf{A})^{-1}\| = \left\|\sum(\mathbf{A}^+\Delta\mathbf{A})^k\right\| \leq \sum \|(\mathbf{A}^+\Delta\mathbf{A})\|^k = \frac{1}{1 - \|\mathbf{A}^+\Delta\mathbf{A}\|}.$$

Hence,

$$\begin{aligned} \mathbf{x}^* - \mathbf{x} &= (\mathbf{A} + \Delta\mathbf{A})^+(\mathbf{b} + \delta\mathbf{b}) - \mathbf{A}^+\mathbf{b} \\ &= (\mathbf{I} + A^+\Delta\mathbf{A})^{-1}\mathbf{A}^+(\mathbf{b} + \delta\mathbf{b}) - \mathbf{A}^+\mathbf{b}; \end{aligned}$$

therefore

$$\begin{aligned} (\mathbf{I} + \mathbf{A}^+\Delta\mathbf{A})(\mathbf{x}^* - \mathbf{x}) &= \mathbf{A}^+(\mathbf{b} + \delta\mathbf{b}) - \mathbf{A}^+\mathbf{b} - \mathbf{A}^+\Delta\mathbf{A}\mathbf{A}^+\mathbf{b} \\ &= \mathbf{A}^+(\delta\mathbf{b} - \Delta\mathbf{A}\mathbf{x}); \end{aligned}$$

and then

$$\begin{aligned}\frac{\|\mathbf{x}^* - \mathbf{x}\|}{\|\mathbf{x}\|} &\leq \frac{\|\mathbf{A}^+\|}{1 - \|\mathbf{A}^+\Delta\mathbf{A}\|}\frac{\|\delta\mathbf{b}\| + \|\Delta\mathbf{A}\mathbf{x}\|}{\|\mathbf{x}\|} \\ &= \frac{\mathrm{cond}(\mathbf{A})}{1 - \|\mathbf{A}^+\Delta\mathbf{A}\|}\frac{\|\delta\mathbf{b}\| + \|\Delta\mathbf{A}\mathbf{x}\|}{\|\mathbf{A}\|\|\mathbf{x}\|} \\ &\leq \frac{\mathrm{cond}(\mathbf{A})}{1 - \|\mathbf{A}^+\Delta\mathbf{A}\|}\left(\frac{\|\delta\mathbf{b}\|}{\|\mathbf{A}\mathbf{x}\|} + \frac{\|\Delta\mathbf{A}\|\|\mathbf{x}\|}{\|\mathbf{A}\|\|\mathbf{x}\|}\right) \\ &\leq \frac{\mathrm{cond}(\mathbf{A})}{1 - \|\mathbf{A}^+\Delta\mathbf{A}\|}\left(\frac{\|\delta\mathbf{b}\|}{\|\mathbf{b}\|} + \frac{\|\Delta\mathbf{A}\|}{\|\mathbf{A}\|}\right).\end{aligned}$$

□

Part II

Accelerating realistic image synthesis

4 Boosting Monte Carlo Renderers

This chapter describes a new multi-scale filter for accelerating Monte Carlo renderers. Each pixel in the image is characterized by the colors of the rays that reach its surface. The proposed filter uses a statistical distance to compare the ray color distribution associated with each pixel at each scale. Based on this distance, it decides whether two pixels can share their rays. This simple and easily reproducible algorithm provides a PSNR gain of 10 to 15 decibels, or equivalently accelerates the rendering process by using 10 to 30 times fewer samples without observable bias. The algorithm is consistent, does not assume a particular noise model, and is immediately extendable to synthetic movies. Being based on the ray color values only, it can be combined with all rendering effects.

4.1 Introduction

In computer graphics, producing high quality realistic images in a reasonable amount of time is still a major challenge. The goal of a global illumination algorithm is to estimate the light distribution in a scene. The color of each pixel in the image results from the superposition of light rays transported by an infinite number of paths that lead to it, either directly from light sources, or indirectly after bouncing in the scene. The light distribution in a scene can be obtained as a solution of the rendering equation, an integral equation that models the radiance equilibrium as a light transport in a scene [Kajiya 1986]. Solving this equation for real scenes is an intractable problem. Approximate solutions are usually obtained by Monte Carlo numerical integration techniques where image pixels are formed by averaging the contribution of stochastic rays cast from the camera through the scene. The main limitation of Monte Carlo rendering is that the variance of the estimator decreases linearly with the number of stochastic samples. Thus the root mean squared error of the estimated image decreases as the square root of the number of primary rays cast from the camera (which we call samples from now on). While variance reduction techniques such as importance sampling, Russian roulette, or Markov Chain Monte Carlo methods can be used to accelerate convergence, still several hours or even days may be necessary to produce noiseless photorealistic images.

To reduce the time required by Monte Carlo rendering to produce good quality images, two main strategies have been proposed, that may be called *adaptive rendering* and *rendering post-processing*. In the first strategy, the idea is to act during the rendering process by locally adapting the number of rays cast per pixel, depending on the complexity of certain zones. Post-processing is applied once rendering has been completed, and mainly consists of filtering or interpolating either samples or pixels. Both strategies can be combined.

It is worth noting that the target quality of the images may vary depending on the application.

The quality required for pre-visualization, where important time constraints have to be met, is clearly not as high as for applications where photo-realistically rendered scenes are an objective by themselves. In pre-visualization scenarios, computational time reduction is obtained by using the renderer to produce only a very small number of samples (say, 2 to 8 samples per pixel). In order to produce images of high enough quality from such a sparse and noisy data, it is necessary to filter or re-synthesize samples using as much information as possible. Indeed, for each sample the rendering system keeps track of relevant information associated to the ray path: geometric, color and texture features, object and material properties, Monte Carlo random parameters, etc. Using these *fat samples*, state of the art methods such as the ones proposed by Sen and Darabi [2012] or by Lehtinen et al. [2012] produce spectacular results. However, the quality of the results obtained by this approach remains scene dependent, being potentially affected by the strong under-sampling of high dimensional data. The larger the number of effects that are simultaneously present, the higher the risks of this under-sampling. Proper up-sampling or interpolation of the sample space is therefore only possible under strong regularity conditions on the fat samples distribution. This explains why the best performances are observed for highly diffusive scenes (where impressive results are obtained from only one sample per pixel). As pointed out by Lehtinen et al. [2012], poor performance is instead expected when the scene contains high frequency illumination effects, incompatible with a low sampling rate.

In short, the generation of high quality images, specially when simulating complex effects such as anti-aliasing, indirect illumination, depth-of-field, motion, requires a large number of rendered paths to correctly sample the path space. The required number of fat samples is certainly too large, not only because of the computational time that would be required to process them but, most fundamentally, because its memory storage would exceed any reasonable capacity limit (more than 100 bytes per fat sample [Sen and Darabi 2012; Lehtinen et al. 2012]). The natural alternative is to give up using fat samples, and to store only part of their information. In the limit, the information can be reduced to *color samples*, that is the final color transported by each ray when hitting the screen. In this case, we say that the method works on the *screen space*, as opposed to the previous methods which work in the space of paths. Working on the screen space allows one to avoid memory saturation, while keeping a number of samples which may be large enough to capture the sample space variability. The works by Rousselle et al. [2011], Dammertz et al. [2010] and Overbeck et al. [2009] are among the most representative ones of this kind of approach.

In the present work we propose and study a new, intermediate, filtering approach that works on the screen space but keeps and uses the color samples at each pixel. Thus, it can be coupled with any Monte Carlo renderer keeping a record of the samples color. The cornerstone of the proposed method is to measure the similarity between any two pixels as the statistical distance between the histograms of rays color that hit them. If the comparison is positive, the ray color histograms of the similar pixels can be fused. The final color of a pixel is then obtained as the average of a all rays color of all similar pixels. This fusion is made still more reliable by comparing patches instead of pixels, and by allowing long range interaction by a multiscale procedure. The ray color histogram characterizes better the physical and geometric properties of a pixel than just its color or the color of its neighbors. The proposed approach is related to bilateral filters, which were first applied to denoise Monte Carlo rendered scenes by Xu and Pattanaik [2005]. The idea of comparing patches instead of individual pixels goes back to Buades et al.'s Non-Local Means 2005. Our approach still presents a fundamental difference with Non-Local Means or any classic variation of bilateral filters proposed in image processing or computer graphics: instead of defining similarity by computing distances between pixels color, we compute distances between color distributions and fuse them when the distance is small enough. For this reason, we will call this method *ray histogram fusion*

RHF. Distances based on distributions are of course much more informative than comparing just their averages, as bilateral filters do.

RHF is simple, easy to implement, and therefore fully reproducible. It is independent of the sample generation process. It can be easily coupled with any renderer and even with any other acceleration method. Most importantly, the method does not make any particular assumption on the scene. As will be demonstrated by our experiments, it therefore copes with a wide range of scenes and multiple simultaneous effects. Finally, its time and memory complexities grow linearly with the image size and are independent of the number of input samples.

The limitations of RHF are also clear. Its performance depends on the degree of self similarity of the scene, which fortunately is usually high [Lebrun et al. 2012], and the price to pay for its generality is the requirement of a relatively large number of input samples.

The plan of the chapter is as follows. In Section 4.2 we review relevant previous work. Section 4.3 defines a pixel similarity measure based on the corresponding cast rays color, and discusses its statistical interpretation. Section 4.4 describes the RHF algorithm, and shows how it successfully makes use of the whole ray color histogram information. Section 4.5 reports quantitative and qualitative results on the algorithm performance. We close with Section 4.6, discussing limitations of our approach and outlining future work and conclude in Section 4.7.

4.2 Previous Work

A thorough analysis of Monte Carlo rendering is far beyond the scope of the present work. The interested reader may consult the introductory book by Dutré et al. [2006] and the one by Pharr and Humphreys [2010]. However, for what follows it is enough to note that there are mainly two approaches to reduce the time required by Monte Carlo rendering to obtain good quality images.

One of these approaches is *adaptive sampling*. This class of algorithms locally adapt the number of rays cast per pixel. The idea is to increase the number of samples in complex parts of the scene while maintaining a reduced number in simple parts, such as flat regions. Complex textures or defocused zones are typical elements that require large amounts of rays to be properly rendered. Hachisuka et al. [2008] (MDAS) proposed to adaptively distribute a set of samples in the full, multidimensional sampling domain where the rendering equation is computed. However, as more Monte Carlo effects are considered (e.g. depth of field, motion blur, area lighting, etc.) the dimension of this space will be larger and thus will suffer from the curse of dimensionality. One of the most significant adaptive sampling algorithms is certainly the Adaptive Wavelet Rendering (AWR) by Overbeck et al. [2009]. This method adaptively distributes Monte Carlo samples in the screen space to reduce the variance of a wavelet basis scale coefficients. Then, the image is reconstructed from these non-uniformly distributed samples by using a suitable wavelet approximation.

Soler et al. [2009] proposed to analyze the depth of field effect in the Fourier domain. By properly predicting the local bandwidth their algorithm adaptively samples the multidimensional domain. In a similar fashion, Egan et al. 2009; 2011b; 2011a addressed motion blur, soft shadows and directional occlusions respectively, by adaptively sampling the multidimensional domain followed by a sheared reconstruction. This allowed reusing samples between pixels in specific effects.

The reconstruction scheme proposed by Rousselle et al. [2011] attempts to minimize the mean squared error. The idea is that given the current distribution of samples in the screen space, the algorithm chooses the best reconstruction filter (among a set of predefined filters - e.g. Gaussian filters) at each pixel to minimize some error criterion. Next, given the current filter selection, new samples are distributed to minimize the error. Thus, this algorithm is both an adaptive sampling and reconstruction filtering. This state of the art algorithm will be used in the experimental section for comparison.

The other approach is *denoising* or *adaptive filtering*. In this family of algorithms, the existing set of samples are combined to produce a better estimator of the pixel color using sample information in a pixel. Adaptive filtering may take place at sample level (i.e., primarily filtering samples) or at pixel level (i.e., primarily filtering pixel values). The majority of these methods can actually be written as generalizations of the *bilateral filter* [Paris et al. 2007] applying a weighted average of the samples (resp. of the pixels) in a neighborhood. The complexity of the method depends on whether it is applied at a pixel or sample level and how deep the method digs into the rendering information (e.g., information about each sample history: color, normal, object of the last impact; information about the random parameters used to calculate the sample) in order to compute the weights of the samples. In order to show how the conception of the proposed filter appears as the natural evolution of the previous work, we briefly present significant contributions, from a historical perspective. We will see that the general trend in this evolution is to rely more strongly in the auxiliary information available from the rendering system.

The simplest adaptive filters act at pixel level, like any filter used in classical image processing. Lee and Redner [1990] presented a seminal work defining an alpha trimmed filter (a generalization of the median/mean). Jensen and Christensen [1995] proposed to apply Gaussian or median filters with 3×3 pixels support to light having been reflected diffusely at least twice. The trilateral filter of Choudhury and Tumblin [2003] involves an adaptive neighborhood function and the image gradient. Again a classic image bilateral filter was proposed by Xu and Pattanaik [2005]. Notice that unlike the work by Lee and Redner [1990], classical bilateral filters cannot remove outliers. To overcome this limitation, the weights of the bilateral filter by Xu and Pattanaik are computed based on a denoised version of the original image.

More complex filters make use of sample information available from the renderer in order to filter still at a pixel level. Rushmeier and Ward [1994] proposed to spread out noisy pixels (e.g. pixels whose variance is larger than a threshold after a fixed number of iterations) into a region of influence. A noisy pixel will contribute to several denoised output pixels, and since the filter is normalized no energy will be leaked. McCool [1999] proposed another classical filter that uses pixels geometric information. It is an anisotropic diffusion (of the Perona-Malik type) removing noise from Monte Carlo rendering. The conductance function that models the strength of the diffusion scheme in a pixel is estimated from a coherence map using depth and normal information gathered during rendering (contained in the G-buffer) along with a color coherence map. More recently Dammertz et al. [2010] presented a fast wavelet filtering scheme designed for ray traced Monte Carlo global illumination images. For that purpose the filter uses RT-buffer information about direct or indirect illumination, and the buffer information on normals and position. The bilateral filter is also invoked by Xu et al. [2011] to denoise images created with complex light paths in smoke or fog. In this work, additional bilateral weights based on the path gradient direction are used to better guide the denoising scheme.

The last class of filters uses the additional sample information to adaptively filter the sample values. Shirley et al. [2011] addressed the question of noise in defocused or motion blurred regions. The image filter is adapted to the *a priori* knowledge of the kind of blur in a given image region. This is a very natural and successful *ad hoc* strategy for these regions. Probably the most impressive results are those recently reported by Sen and Darabi [2012]. This method uses the whole information of the rendering process and the whole information on each numerical photon to denoise by bilateral filtering. The bilateral filter takes simultaneously into account in its weights the sample position and spatial neighborhood in the image, the random synthesis parameters, the scene features (normal, world space position, texture values) and finally the sample color. It computes as a mutual information the statistical dependence on the random generation parameters of the pixels sharing the same features and colors. Although the results are outstanding at very low samples per pixel, the complexity of the method makes it not scalable to generate high quality images from a large number of input samples.

Lehtinen et al. [2011] described a reconstruction technique that allows rendering a combination of motion blur, depth of field and soft shadows by exploiting the anisotropy in the temporal light field. The effective sampling rate is increased by a large factor by efficiently reusing samples between pixels. Recently Lehtinen et al. [2012] generalized these ideas to deal with indirect illumination. By contemplating the properties of diffuse surfaces, their algorithm permits to interpolate the light field to produce results similar to those that would have been obtained by rendering a much larger number of samples. For instance, from an input image of 8 samples per pixel, they synthesize images of 256 samples per pixel, whose quality is similar to 512 samples per pixel generated by standard path tracing. While the quality increase is impressive, the noise level in these images is still too strong for applications requiring high quality images. Our algorithm is somehow complementary to this approach. Indeed, it can be used to boost the performance of a pure Monte Carlo renderer or any other set of samples like the ones generated by Lehtinen et al. [2012], and can perfectly deal with a number of samples in this order of magnitude.

The above bibliographical analysis has shown that most Monte Carlo denoising methods are generalizations of the bilateral filter (or sigma-filter [Lee 1983]). The general principle behind the bilateral filter is that similar pixels must be denoised jointly, being different samples of the same model. This is also implicitly used by the sigma-filter and by the NL-means algorithm [Buades et al. 2005]. In computer graphics, ray information permits to identify still more rigorously than in classic image processing the pixels sharing the same model. Indeed, all ray samples hitting a given pixel and its neighbors can be used for that purpose.

4.3 Proposed Approach

4.3.1 Rationale

In contrast to classical photography where only the energy arriving at the sensor plane can be measured, in a rendering scenario much more information about pixel formation is available. In particular, the light contribution and the screen position of each path can be stored, as well as the associated geometrical and scene information about the objects encountered along the ray path.

As pointed out by Veach [1997], the light transport problem can be stated in the space of paths, and global illumination can be estimated by computing a transport measure over each individual path. Under this *path integral formulation*, each pixel color $u(\mathbf{x}) = (u_R(\mathbf{x}), u_G(\mathbf{x}), u_B(\mathbf{x}))$ is given by the integral over all possible light paths

$$u(\mathbf{x}) = \int_{\Omega_{\mathbf{x}}} f(\mathbf{p})d\mu(\mathbf{p}),$$

where $\Omega_{\mathbf{x}}$ is the space of paths originated at pixel $\mathbf{x}$, $\mathbf{p}$ is a path of any length, and $d\mu(\mathbf{p})$ is a measure in the path-space. The function $f(\mathbf{p})$ describes the energy contribution through a path $\mathbf{p}$ and is the product of several scene factors due to the interaction of light within the path plus initial self-emitted radiance and importance distributions. Thanks to this formulation, the image color at pixel $\mathbf{x}$ can be estimated from $n_{\mathbf{x}}$ random paths $p_{\mathbf{x}}^1, \dots, p_{\mathbf{x}}^{n_{\mathbf{x}}}$, generated by an appropriate Monte

Carlo sampling procedure. If $c_{\mathbf{x}}^{j}$ denotes the color transported by random path $p_{\mathbf{x}}^{j}$ (for instance, in path tracing $c_{\mathbf{x}}^{j} = f(p_{\mathbf{x}}^{j})$), the Monte Carlo approximation of $u(\mathbf{x})$ is computed as

$$\tilde{u}(\mathbf{x}) = \frac{1}{n_{\mathbf{x}}} \sum_{j=1}^{n_{\mathbf{x}}} c_{\mathbf{x}}^{j}. \tag{4.1}$$

Consider now the Monte Carlo approximation error $n(\mathbf{x})$, given by

$$n(\mathbf{x}) = \tilde{u}(\mathbf{x}) - u(\mathbf{x}). \tag{4.2}$$

The Monte Carlo approximation is asymptotically unbiased, but the mean squared error $E[n^2(\mathbf{x})]$ decays linearly with the number of samples $n_{\mathbf{x}}$. Consequently, unless the rendering system spends several hours or even days producing samples, the resulting images will be contaminated by white noise. This is a consequence of the fact that the Monte Carlo samples are independent and therefore the random process $\{n(\mathbf{x}), \mathbf{x} \text{ image pixels}\}$ is white.

One possibility to reduce the approximation error while keeping the rendering time reasonable is to render fewer samples, and to filter the pixel values afterwards. Filtering will always result in a significant variance reduction, however, it may also severely increase the approximation bias. The only filtering processes that do not introduce bias are those that combine pixels of the same "nature", that is pixels $\mathbf{x}$ having the same ideal value $u(\mathbf{x})$. While identifying two similar pixels $\mathbf{x}$ and $\mathbf{y}$ based on the unknown pixel values $u(\mathbf{x})$ and $u(\mathbf{y})$ is of course impossible, it is reasonable to expect that their samples color $\{c_{\mathbf{x}}^{1}, \dots, c_{\mathbf{x}}^{n_{\mathbf{x}}}\}$ and $\{c_{\mathbf{y}}^{1}, \dots, c_{\mathbf{y}}^{n_{\mathbf{y}}}\}$ will follow similar distributions. Moreover, if N pixels share the same sample color distribution, the union of the samples can be seen as an N times larger super-set following the underlying distribution. By simply averaging them the variance reduction is increased by a factor of N.

The cornerstone of the proposed approach is to find the most similar pixels to each given pixel by comparing their underlying sample color distributions. This is the object of the next section.

4.3.2 Distribution-driven pixel similarity

Consider the empirical distribution of the samples color at a given pixel. Figure 4.1 depicts this distribution for five different pixels on two different scenes, for samples generated by a Monte Carlo path-tracing algorithm. In the first example (top row) the three pixels were selected because their colors are extremely close. A quick visual inspection shows immediately that the samples of the two edge pixels follow roughly the same color distribution, and that this distribution is considerably different from the one of the third pixel. This example illustrates to what extent the information provided by the sample color distribution can help discriminate pixels of different nature, even when their pixels color are similar.

In the following, we denote by $\mathcal{C}_{\mathbf{x}} = \{c_{\mathbf{x}}^{1}, \dots, c_{\mathbf{x}}^{n_{\mathbf{x}}}\}$ the set of the color of samples cast from pixel $\mathbf{x}$, and by $h(\mathbf{x})$ the corresponding empirical color distribution. To measure pixel similarity we propose to use the binned empirical distributions as pixel descriptors. Since in general we deal with tri-stimulus color images, we can choose to build this descriptor either as a single histogram in the three-dimensional color space, or as three one-dimensional histograms (one per color channel).

Given the samples color $\mathcal{C}_{\mathbf{x}}$ and $\mathcal{C}_{\mathbf{y}}$ at pixels $\mathbf{x}$ and $\mathbf{y}$, and their corresponding n_b-binned distributions $h(\mathbf{x}) = (h_1(\mathbf{x}), \dots, h_{n_b}(\mathbf{x}))$ and $h(\mathbf{y}) = (h_1(\mathbf{y}), \dots, h_{n_b}(\mathbf{y}))$, the Chi-Square distance is given by

$$d_{\chi^2}(\mathcal{C}_{\mathbf{x}}, \mathcal{C}_{\mathbf{y}}) = \sum_{k=1}^{n_b} \frac{\left(\sqrt{\frac{n_{\mathbf{y}}}{n_{\mathbf{x}}}} h_k(\mathbf{x}) - \sqrt{\frac{n_{\mathbf{x}}}{n_{\mathbf{y}}}} h_k(\mathbf{y}) \right)^2}{h_k(\mathbf{x}) + h_k(\mathbf{y})}, \tag{4.3}$$

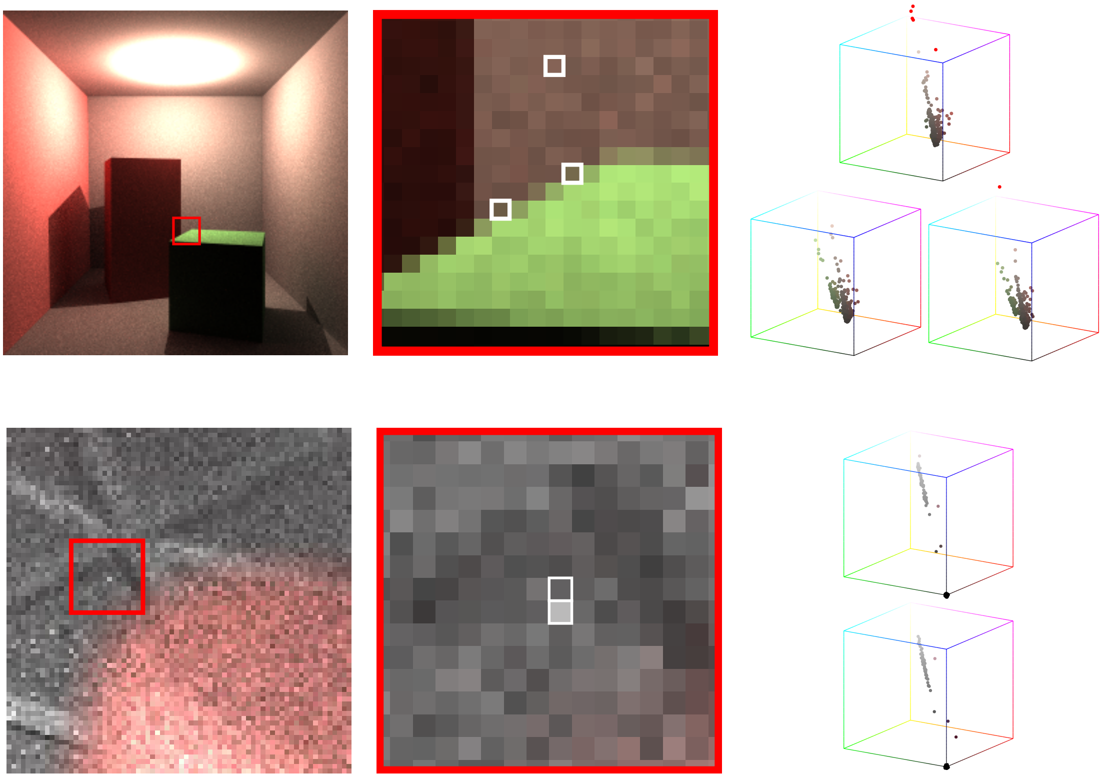


Figure 4.1: The top row singles out three pixels in Cornell Box scene and their sample color distributions. (The samples with color values falling out of the $[0,1]^3-$box are by convention colored in red.) The first pixel, situated on the brown wall, has a unimodal sample color distribution. The other two pixels belong to an occlusion boundary showing a bimodal green-brown distribution. This feature is shared by many pixels on the same boundary, which can therefore share their samples. The bottom row shows two pixels of the `toasters` scene with different colors. Their sample color distributions are nevertheless very similar and will therefore be merged as well.

where $n_{\mathbf{x}} = \sum_k h_k(\mathbf{x})$ and $n_{\mathbf{y}} = \sum_k h_k(\mathbf{y})$ are the total number of samples in each set.

In order to take into account spatial coherence, the previous pixel-wise distance can be extended to patches of half-size w centered at $\mathbf{x}$ and $\mathbf{y}$ as follows,

$$d_{\chi^2}(P_{\mathbf{x}}, P_{\mathbf{y}}) = \sum_{|\mathbf{t}|\leq w} d_{\chi^2}(\mathcal{C}_{\mathbf{x}+\mathbf{t}}, \mathcal{C}_{\mathbf{y}+\mathbf{t}}). \tag{4.4}$$

Since the order in which the samples are calculated is irrelevant, the sample color distribution appears as a natural and complete descriptor of the compared sets. There are different ways of measuring the similarity between two distributions depending on the data type. In the case of continuous data, the Cramer-von Mises [Anderson 1962; Anderson and Darling 1952], the Kolmogorov-Smirnov [Stephens 1970; Press et al. 2007] or the Kantorovich-Mallows-Monge-Wasserstein distances (also known as the Earth Mover's Distance [Rubner et al. 1998]) are all accepted ways to compare distributions. These three similarity measures are computed as L^p distances between the two cumulative distributions (L^∞, L^2 and L^1 respectively). For categorical data, the most popular measure to compare distributions is the χ^2 distance previously defined in (4.3).

By discretizing the data in a fixed number of histogram bins, the computational complexity of measuring the similarity between two data sets can be kept bounded and independent of the number of samples. This is important since this similarity measure is evaluated a large number of times. Thus, the color space will be divided into fixed bins, and the χ^2 distance fits well to this form for the data. However, if an image is rendered with very few samples, one of the other two metrics would be preferable.

Since we are only interested in similar distributions, we set a threshold κ on the Chi-Square distance divided by the number of non-empty bins k,

$$\kappa = {}^{d_{\min}}\!/_{k}.$$

The normalization by the number of non-empty bins is necessary since only the bins carrying information should be considered in the comparison.

Remark: comparing pixel values versus comparing distributions State of the art image denoising algorithms measure pixel similarity by comparing pixel colors. Indeed, the bilateral filter and NL- Means replace each noisy pixel by a weighted average of the most similar ones. In the case of NL-means, the pixel comparison is performed with patches centered around each pixel. For a very recent review on patch based denoising methods, we refer to [Lebrun et al. 2012] and for a fast implementation to [Adams et al. 2009].

Nevertheless, image denoising algorithms must know or measure the noise variance to evaluate properly the similarity of noisy samples. Fortunately, Monte Carlo rendering is an almost ideal situation where mean and variance values of the rays cast from each pixel can be estimated.

The main disadvantage of this formulation is that it cannot distinguish noise from intrinsic pixel variability. As a first example, suppose that a pixel is situated on an edge. In that case the sample color distribution will be at least bi-modal. Thus, it will probably have a large variance. This variance will result in a large tolerance to differences in the means, and consequently different pixel types may be wrongly mixed up. A case of this type is shown in Figure 4.1 (top row).

On the other hand, by directly comparing distributions, pixels lit from several sources can be better clustered. In the case of the histogram comparison, we will need no implicit nor explicit noise model assumption.

The bottom row of Figure 4.1 shows two pixels with very different pixel values. This is the consequence of the presence of a single very bright ray sample in one of the distributions. By comparing the ray color distributions, it is nevertheless possible to conclude that both pixels are from the same “nature”, while this conclusion could not be reached by comparing the averages.

4.3.3 Distribution-driven average

For each pixel $\mathbf{x}$, we define $\mathcal{N}_\epsilon(\mathbf{x})$ as the set of pixels $\mathbf{y}$ whose centered patches $P_\mathbf{y}$ are such that $d_{\chi^2}(P_\mathbf{x}, P_\mathbf{y}) \leq \epsilon$. Then, if ϵ is such that these pixels are of the same nature as $\mathbf{x}$, the maximum likelihood estimator of the noiseless pixel color is simply their arithmetic mean

$$\bar{u}(\mathbf{x}) = \frac{1}{|\mathcal{N}\epsilon(\mathbf{x})|} \sum_{\mathbf{y} \in \mathcal{N}_\epsilon(\mathbf{x})} \tilde{u}(\mathbf{y}).$$

Unlike the previous estimator, where only the center of the patch is averaged, we can proceed to denoise the whole patch, and to denoise the image patchwise. Let us denote by $V_\mathbf{x}$ the color values of a denoised patch centered at pixel $\mathbf{x}$. Similarly, this denoised patch can be computed by averaging the patches which are at a Chi-square distance smaller than ϵ:

$$V_\mathbf{x} = \frac{1}{|\mathcal{N}\epsilon(\mathbf{x})|} \sum_{\mathbf{y} \in \mathcal{N}_\epsilon(\mathbf{x})} \tilde{u}(P_\mathbf{y}),$$

where we use the convention that $\tilde{u}(P_\mathbf{y})$ is the evaluation of u on each pixel in patch $P_\mathbf{y}$.

In this way, by applying this aggregation procedure for all patches in the image, we shall dispose of $(2w+1)^2$ possible estimates for each pixel. These estimates can be finally averaged at each pixel location in order to build the final denoised image.

$$\tilde{u}(\mathbf{x}) = \frac{1}{(2w+1)^2} \sum_{|\mathbf{y}-\mathbf{x}| \leq w} V_\mathbf{y}(\mathbf{y}-\mathbf{x}).$$

This patchwise implementation is the one considered in this work.

4.3.4 Removing low-frequency noise

As already mentioned, in a pure Monte Carlo scenario the approximation error is characterized by a white random noise. This means that all frequencies are equally contaminated by noise. The proposed filtering procedure described so far filters noise at patch scale. Long wavelength noise cannot be eliminated by this procedure, because long wavelength structures cannot be captured by small patches. Removing noise at lower frequencies requires a (straightforward) multi-scale extension of the method. Let us define two useful operators, the $s\times$ Gaussian downsampling $D_s u(\mathbf{x}) := (G_{2^s\sigma} * u)(2^s\mathbf{x})$ and U_s the $s\times$ bicubic interpolator.

Now, for each scale s, the corresponding histograms $h^s(\mathbf{x})$ have to be computed. Since each pixel at scale s results from the fusion of a set of neighboring pixels in the original finer scale, the new histograms are obtained by fusing the color histograms of all pixels in the same neighborhood. To obtain $h^s(\mathbf{x})$, the same down-sampling operator D_s can be applied to the original color distribution $h(\mathbf{x})$. Then, at each scale, the resulting histograms are re-normalized so that the sum of their areas is preserved across scales (thus preserving the original total number of samples in the finer scale).

Given a noisy image input $\tilde{u}$ and its respective pixel color distribution $h(\mathbf{x})$ the multi-scale histogram fusion proceeds as follows:

1. Generate the Gaussian multi-scale sequence: $\tilde{u}^0 = \tilde{u}$, $\tilde{u}^s = D_s\tilde{u}$, $s = 1, \dots, N$, and their respective sample color distributions.

2. Apply the denoising algorithm separately to each scale to recover $\bar{u}^0, \bar{u}^1, \dots, \bar{u}^N$.

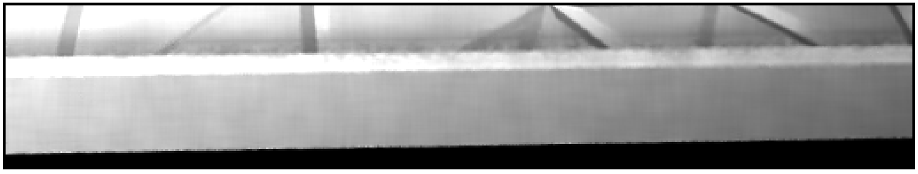
3. Compute the final image $\bar{u} = \hat{u}_0$ by the recursion

$$\hat{u}_i = \bar{u}_i - U_1 D_1 \bar{u}_i + U_1 \hat{u}_{i+1}$$

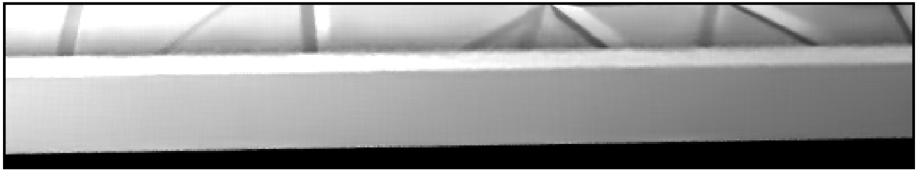
initialized with $\hat{u}_N = \bar{u}_N$ for $i = N$.

Figure 4.2 shows the importance of dealing with noise at multiple scales. When filtering only at a single fine scale, conspicuous low frequency noise remains. This noise is almost completely eliminated by the multi-scale procedure with three scales.

noisy input



single-scale denoised



three-scale denoised

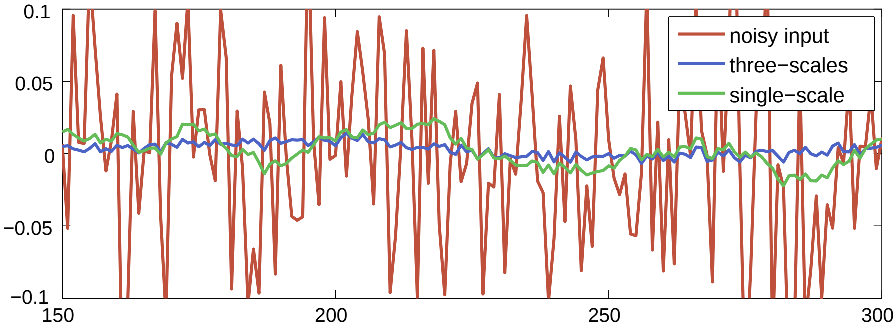


Figure 4.2: The multi-scale approach eliminates low-frequency noise, as can be seen in the second and third row, and in the profile for a particular line shown in the bottom row.

4.4 Implementation Details

As previously stated, our approach builds on two basic blocks: the estimation of the sample color distribution at each pixel, and a non-local multi-scale filtering based on averaging pixels sharing similar sample color distributions. This requires two kinds of data from the rendering system: the noisy Monte Carlo image $\tilde{u}(\mathbf{x})$ and the associated sample color histograms $h(\mathbf{x})$.

A fundamental aspect of our method is that sample color histograms can be computed on the fly, in parallel with the Monte Carlo rendering process. This is extremely important, since it makes the memory requirements independent from the number of rendered samples. The memory complexity bounds are fixed by the number of pixels, and therefore, as pointed out in the introduction, the input Monte Carlo images may consist of a large enough number of samples to produce high quality images.

This section gives the implementation details to estimate the sample color distribution and to perform the non-local multi-scale filtering.

4.4.1 Computing the color distribution of samples

To approximate a distribution using a histogram, one has to divide the range of possible values into discrete bins and count the number of samples within each bin. Smoother estimates can be obtained using kernel density estimation, by interpolating the contribution of each sample using a kernel. In this work, we used a triangular kernel to linearly interpolate the contribution of each sample color value to its adjacent bins.

We recall that pixel values are obtained by averaging sample color values. Hence, despite the saturation value for pixels (perfect white) is one, the range spanned by the sample values is much larger. In order to take into account the fact that high-energy (bright) samples are less frequent than low-energy ones, the bins are designed so that their sizes increase with the sample value, following an exponential law of exponent 2.2. The range covered by the histograms is set to $[0, 7.5]$, and all samples exceeding this range are counted assigned to the last bin. It is worth mentioning that although histogram comparison is not particularly sensitive to these parameters, they must be chosen to cover the dynamic range adequately.

In general, sample values have a tri-stimulus color representation. Therefore we can either compute one three-dimensional distribution (3D) where bins are boxes in the full 3D color space, or compute three one dimensional distributions, one for each color. Although distributions in the 3D color space can capture inter-color correlations, a much larger number of bins are required to keep the same quantization step, and consequently a larger number of samples. In Section 4.5 we present a comparison of both strategies. This comparison shows that there is no advantage in using the full 3D color space.

The estimation of the color distribution can be done on the fly while samples are being computed. This keeps the memory bounded and independent of the number of samples per pixel produced by the rendering system.

4.4.2 Filtering

The implementation of the RHF filter is straightforward. In addition to the parameters needed to compute the histogram, four parameters are involved in the algorithm: the number of scales n_s, half the patch size w, half the search block size b, and the χ^2 distance threshold.

The search of similar patches is restricted to a block of size $(2b+1) \times (2b+1)$. This is reasonable since the probability that two patches are similar will be smaller if one is distant from the other. The threshold is directly set on the Chi-square distance as a product of κ (the user parameter) by the number of bins k where both histograms are non-empty. A pseudo-code of both the filtering at each scale and the multi-scale generalization are presented in Algorithms 1 and 2, respectively. In algorithm 1, the denoised version of patch P_i is obtained by averaging all patches Q_j such that $d^2_{\chi}(P_i, Q_j) < k \cdot \kappa$.

Note that the only user parameter is κ. This parameter controls the amount of noise that is removed, or in other words the trade-off between image smoothness and noise. Its optimal choice depends mostly on the samples generation process (the considered renderer). The dependence on the rendered scene is actually very weak, as will be demonstrated by experiments in Section 4.5. A simple intuitive explanation for the dependence of the optimal κ on the rendering method comes from the observation that the value of κ is related to the confidence associated to the color samples. If the samples values are computed with low confidence, the distance threshold should be less restrictive. For instance, in pure Monte Carlo path tracing, each sample carries the energy of a single light path, while in volumetric ray tracing each sample value is computed as the average of several light paths. Therefore, the samples generated with pure path tracing have lower confidence, and this explains why the threshold should be less restrictive.

The practical implications of this fact is that, once a rendering method has been chosen, the value of κ can be safely fixed once for all. Moreover this tuning is not time consuming: indeed, since the distance between patches (the heaviest computational task) is independent of the parameter, its computation can be first performed and then several values of the parameter can be tested with practically no additional cost.

The multi-scale implementation in Algorithm 2, as detailed in Section 4.3.4, sequentially decomposes the input noisy image at each scale, filters each scale using Algorithm 1 and reconstructs the multi-scaled filtered image.

4.4.3 Time complexity

The complexity of the filtering at each scale is $O(Nwbn_b)$ where N is the number of pixels. Note that the computational cost is independent of the number of samples.

In the case that two scales are used the computational cost increases by about 25%, the low-frequency noise filtering being done on a four times smaller image. If n_s scales are used the computational cost is bounded from above by 133% of the filtering time at the finest resolution.

4.5 Experimental Set-up and Results

Different types of scenes containing complex geometries, indirect illumination, depth-of-field and other effects were rendered using the software PBRT-V2 [Pharr and Humphreys 2010]. The color distribution estimation stage was implemented on top of PBRT, so the color histograms were produced online as the samples were computed. The filtering-reconstruction stage was implemented in a stand-alone application which makes use of a multichannel image that to each pixel associates its corresponding sample color histogram, and the noisy Monte Carlo image generated with a box filter.

We compared the proposed algorithm to three different methods, both regarding image quality and execution time. The first one is a pure Monte Carlo rendering (MC): this is the basic approach

Algorithm 1: Single-Scale Ray Histogram Fusion

```
Input: MC image $\tilde{u}$, corresponding histograms $h$, patch size $w$, search block size $b$ , distance
    threshold $\kappa$
Output: Filtered image $\bar{u}$
 1: $\bar{u} \leftarrow 0$
 2: $n \leftarrow 0$                       //auxiliary counter at each pixel in the image
 3: for every pixel $i$ do
 4:     $P_i \leftarrow$ patch centered at pixel $i$
 5:     $W_i \leftarrow$ search block for pixel $i$
 6:     $c \leftarrow 0$ and $V \leftarrow 0$
 7:     for every $j \in W_i$ do
 8:         $Q_j \leftarrow$ patch centered at pixel $j$
 9:         $d \leftarrow$ ChiSquareDistance$(h(P_i), h(Q_j))$
10:         $k \leftarrow$ Number of non-empty bins in $h(P_i) + h(Q_j)$
11:         if $d < \kappa \cdot k$ then
12:             $V \leftarrow V + \tilde{u}(Q_j)$
13:             $c \leftarrow c + 1$
14:         end if
15:     end for
16:     $V \leftarrow V/c$
17:     $n(P_i) \leftarrow n(P_i) + 1$                 // +1 for each pixel in $P_i$
18:     $\bar{u}(P_i) \leftarrow \bar{u}(P_i) + \big(V - \bar{u}(P_i)\big) ./ n(P_i)$
19: end for
```

Notation convention: $\tilde{u}(P_i)$ is the evaluation of $\tilde{u}$ on each pixel in patch P_i (the same applies for $\bar{u}, n, h$). The operator ./ (line 18) represents element-wise division.

Algorithm 2: Ray Histogram Fusion

```
Input: MC image $\tilde{u}$, corresponding histograms $h$, patch size $w$, search block size $b$ , distance
    threshold $\kappa$, number of scales $n_s$.
Output: Filtered image $\bar{u} = \bar{u}_0$
$s \leftarrow n_s - 1$
$n_T \leftarrow \sum_{\mathbf{x},k} h_k(\mathbf{x})$                 // total number of samples
while $s \geq 0$ do
    $u^s \leftarrow D_s(\tilde{u})$
    $h^s \leftarrow D_s(h), \quad n_T^s \leftarrow \sum_{\mathbf{x},k} h_k^s(\mathbf{x}), \quad h^s \leftarrow \frac{n_T}{n_T^s} h^s$
    $\bar{u}_s \leftarrow$ RHF $(u_s, h_s, w, b, \kappa)$
    if $s < n_s - 1$ then
        $\bar{u}_s \leftarrow \bar{u}_s - U_1 D_1 \bar{u}_s + U_1 \bar{u}_{\text{old}}$
    end if
    $\bar{u}_{\text{old}} \leftarrow \bar{u}_s$
    $s \leftarrow s - 1$
end while
```

to generate photorealistic images. This technique is asymptotically unbiased but the variance shows slow (linear) decay with the number of samples.

The second algorithm chosen for comparison is an adaptation of the classic NL-means [Buades et al. 2005]. In image processing, NL-means performs denoising by averaging similar patches. In the rendering scenario, this method is obviously valid and can be improved by considering the noise level at each pixel, estimated from the variance of the samples that are cast from it. The main difference is in the way similar patches are identified. The performance comparison with NL-means will show that the knowledge of the sample color distribution adds a very significant amount of information, not yet contained in the patch colors.

Finally, we also consider comparison with the Adaptive Sampling and Reconstruction technique ASR by Rousselle et al. [2011]. As already discussed in Section 4.2, this Monte Carlo based method can estimate the reconstruction error and control the number of samples cast from each pixel to reduce it. This methods is similar to ours in the sense that it does not rely on fat samples, and uses only the final color of each rendered sample. As such, the method scales well with the number of samples and can be used to produce high quality renderings of complex scenes. ASR is a state of the art algorithm in this class of methods. Comparison is made using the code provided by the authors, and manually setting the parameters to produce the highest possible image quality, while matching the execution time of our algorithm (including both the samples rendering time and the filtering stage).

The success criterion is to get an image that is very close to the ground truth in a much shorter time. Image quality is assessed by comparing results to reference images, generated by pure Monte Carlo rendering with a very large number of samples per pixel. The performance measure is the standard peak-signal-to-noise ratio (PSNR) calculated as $\text{PSNR} = 10 \log \frac{1}{\text{MSE}}$ where MSE is the mean square error to the reference image. The PSNR is a reliable criterion to characterize the quality of the reconstruction. It will nonetheless be complemented by some close-ups of difficult image details. All experiments were performed on a 2× Intel Xeon CPU X5450 @ 3.00GHz (4 cores) with 16GB of RAM.

All the algorithms were run on several scenes from the PBRT software, simulating various effects with varying complexity levels.

In all cases three independent histograms were calculated, one for each channel (R, G, B) with `nbins`=20. The search for similar patches was limited to a 13×13 window centered on the filtered pixel. The κ threshold (the user parameter) was manually set to produce a good balance between smoothness and remaining noise. As previously explained, the optimal value for this trade-off depends mostly on the rendering method. The values of κ that were chosen in the experiments are shown in Table 4.1. Note that for all the renderings performed with a pure path tracing, we set $\kappa = 1$, while for the scene rendered using volumetric ray tracing (`plants-dusk`), $\kappa = 0.37$. This is consistent with the fact that the color samples generated with volumetric ray tracing result from an average of several light paths, and therefore are more precise than in pure path tracing.

A summary indicating all the considered effects, rendering method and image size is shown in Table 4.1.

3D versus $3\times$ 1D Histograms. Table 4.2 illustrates the performance of the method as a function of the number of bins of the rays color histogram. The experiments do not support the use of three-dimensional color space bins. Thus, independent histograms were generated for the R, G, B channels. The number of bins must be large enough to capture the histogram structure, but not too large to grant a robust histogram comparison.

Table 4.1: Summary of the tested scenes.

Scene	Effects	Size	Generation	κ-RHF
`cornell-box`	AI	256×256	path tracing	1.00
`toasters`	AILD	512×512	path tracing	1.00
`plants-dusk`	ALPD	800×400	ray tracing	0.37
`sibenik`	AILD	1024×1024	path tracing	1.00
`yeahright`	AI	800×800	path tracing	1.00
`dragons`	AILDP	512×512	photon mapping + final gathering	0.6

Considered effects: anti-aliasing (A); indirect illumination (I); area lights (L); depth-of-field (D); participating media (P). The scenes `cornell-box`, `plants-dusk` and `sibenik` are from [Pharr and Humphreys 2010] while the scenes `dragons` and `toasters` were taken from [Rousselle et al. 2011].

Table 4.2: Performance comparison: estimating the histogram.

	3D			$3 \times 1D$				
	3^3	4^3	5^3	3×5	3×10	3×15	3×20	3×25
`cornell-box`	40.2	41.0	41.7	39.8	41.6	42.5	43.0	43.3
`toasters`	30.5	31.3	31.7	30.4	32.2	32.8	33.1	33.3
`plants-dusk`	46.2	46.1	46.0	46.2	46.0	45.9	45.8	45.7
`yeahright`	36.0	36.2	36.2	36.3	36.1	36.0	36.0	36.0
`sibenik`	41.4	42.5	43.1	40.3	43.1	43.8	44.2	44.3

Average of the $k = 15$ closest neighbors, the performance metric is the PSNR with respect to a ground truth image. Two different ways of estimating the ray color histogram: bins in the 3D color space or three one-dimensional histograms one for each color. In most cases, taking more bins increases the performance, but also the computational cost of the algorithm. Estimating three independent histograms one for each color gives better results than estimating the histogram in the original 3D color space.

Robustness: Comparing Means versus Comparing Distributions. Suppose that an external Oracle tells us the exact number of closest patches that should be averaged in each pixel to minimize the MSE. By comparing the resulting PSNR, we can compare the maximum theoretical performance of the method. A comparison for different patch sizes is given in Table 4.3. The results show that using histogram information to compare patches permits to significantly reduce the patch size. This is an advantage since it permits the algorithm to be more local, hence better preserving small details. As for the PSNR, the experiment shows that using the color distribution information leads to much better results than using just the pixels colors, as NL-means does.

Comparisons for several scenes. RHF systematically outperforms ASR, as shown in Figures 4.3–4.11. Even if in some scenes both algorithms reach very similar PSNRS, the proposed algorithm does not introduce artifacts while ASR often fails to capture the geometry and causes spots. The PSNR gain by RHF filtering is spectacularly larger than the one that would be obtained by generating more Monte Carlo samples using the same time span. Indeed, a 3db PSNR gain by a pure MC

Table 4.3: Oracle performance comparison.

	NL-means			RHF		
	1×1	3×3	5×5	1×1	3×3	5×5
`cornell-box`	33.4	40.7	44.4	40.1	48.1	49.9
`toasters`	24.0	30.6	34.7	29.8	37.9	40.5
`plants-dusk`	49.0	54.4	55.3	51.4	55.4	55.8
`yeahright`	36.3	43.2	46.0	39.7	46.6	48.2
`sibenik`	34.9	41.5	45.4	40.4	48.6	50.8

For each pixel the ideal number of closest patches to minimize the error with respect to the ground truth image has been computed and fixed. The table compares NL-means and RHF for different patch sizes. It shows that RHF permits to reduce significantly the patch size.

algorithm requires to double the number of samples. Instead, the filtering increases the PSNR by 15 to 20 decibels. This amounts to decreasing the overall sampling time by a factor ranging from 15 to 40.

In the case of the `yeahright` image in Figures 4.12, 4.13 and 4.14 the ASR algorithm produces a slightly better PSNR. This scene is best suited for this algorithm, because several regions are flat and ASR can distribute most of the samples in the problematic parts. Nevertheless, in the shadows RHF produces a more natural smooth result. As previously commented, the NL-means based approach cannot distinguish between a large histogram variance due to pixel complexity, from a variance due to MC noise. This fact is well illustrated in the metal edge of Figure 4.13, which is completely removed by NL-means, while it is well preserved by RHF.

The `plants-dusk` scene with participating media, in Figures 4.15, 4.16 and 4.17 is a very challenging one. Here, the principal problem is the complex geometry of the vegetation. The proposed algorithm tends to blur and to slightly flatten some texture details. Nevertheless, contrarily to ASR, no artifacts are introduced.

To illustrate the fact that the proposed denoising method is independent from the rendering system, in Figures 4.18 and 4.19 we present a filtering experiment that runs on a image generated by photon-mapping and final gathering. This scene comes from [Rousselle et al. 2011]. The noise has been properly removed, and no artifacts are observed.

Extension to animated sequences. The ideas behind this approach can be immediately extended to video sequences where pixels on every frame will be candidates in the search. The supplementary video shows the result of denoising an animated sequence, by implementing this simple generalization. Similar patches are searched within a temporal window of size 3 (namely in the previous, actual and next frame). Although no explicit temporal correlation is enforced, the filtered sequence does not show significant flicker. This is a consequence of the stability of the proposed multi-scale filter.

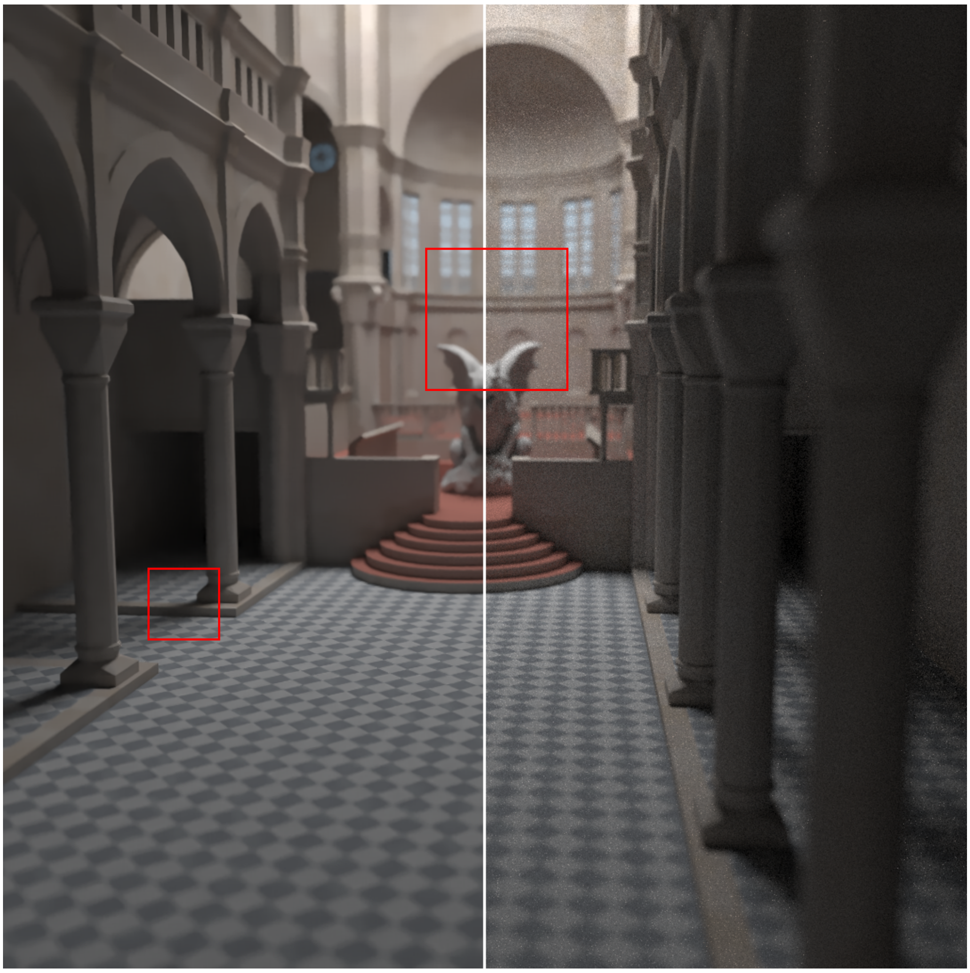
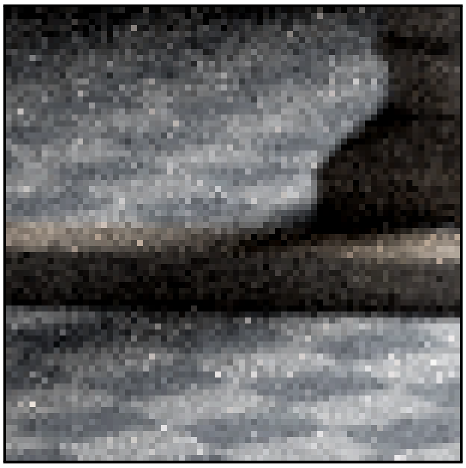
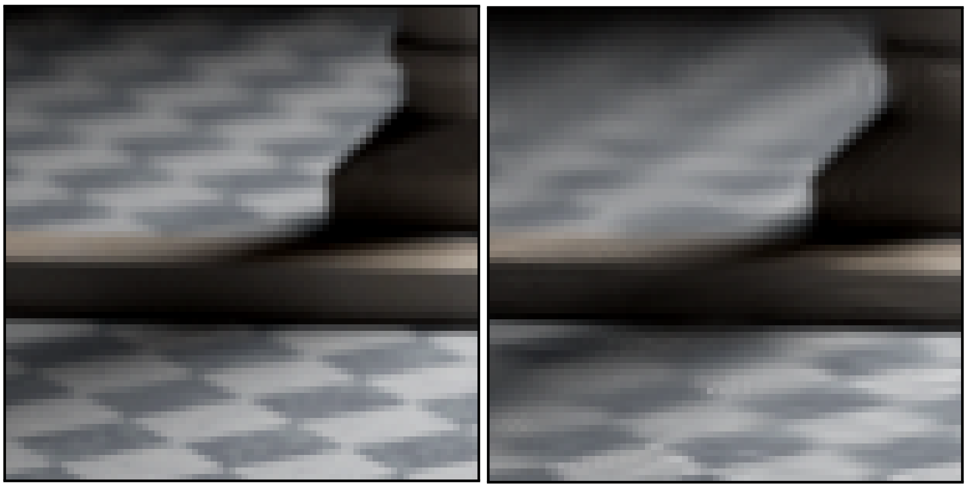


Figure 4.3: Results in `sibenik` scene. The result of `RHF` is shown on the left while the noisy input is shown on the right. For close-ups on difficult parts see Figures 4.4 and 4.5.

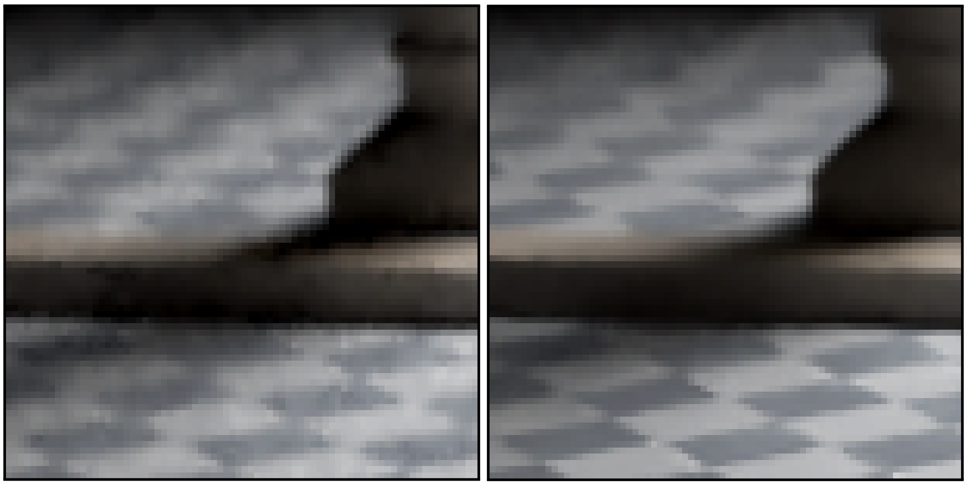


Monte Carlo 128spp 33.7db 307.2s



Reference MC

NL-means 44.1db 327s (20s)



ASR 156spp 44.6db 430.3s

RHF 45.4db 423s (115s)

Figure 4.4: Results in `sibenik` scene (close-ups). In all cases, the PSNR values are given for the whole image. In general ASR tends to create artifacts near edges. RHF produces the best PSNR, with no visible artifacts. For NL-means and RHF the indicated time follows the format *total time (filtering time).*

Monte Carlo 128spp 33.7db 307.2s

Reference MC

NL-means 44.1db 327s (20s)

ASR 156spp 44.6db 430.3s

RHF 45.4db 423s (115s)

Figure 4.5: Results in `sibenik` scene (close-ups). In all cases, the PSNR values are given for the whole image. In general ASR tends to create artifacts near edges. RHF produces the best PSNR, with no visible artifacts. For NL-means and RHF the indicated time follows the format *total time (filtering time).*

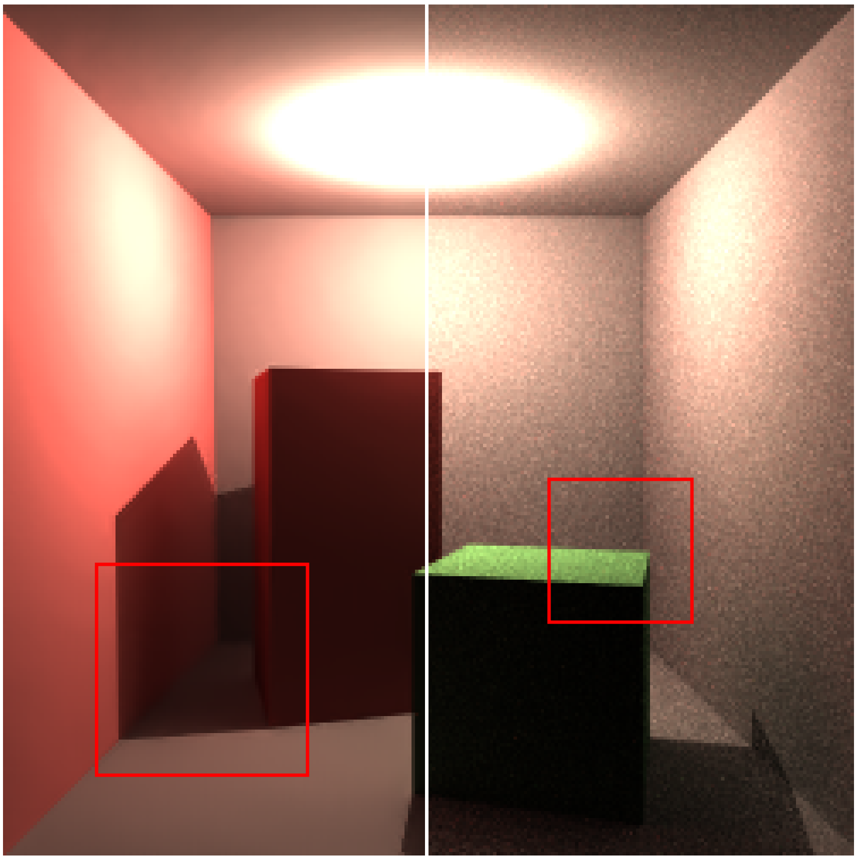
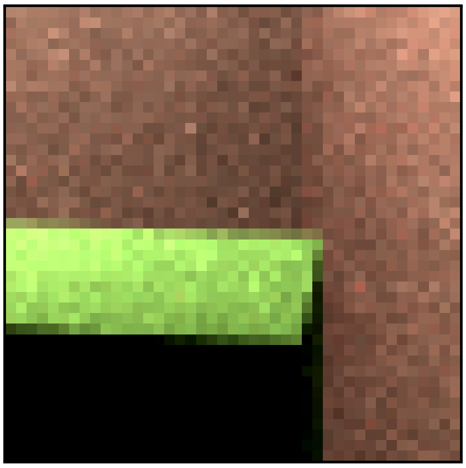
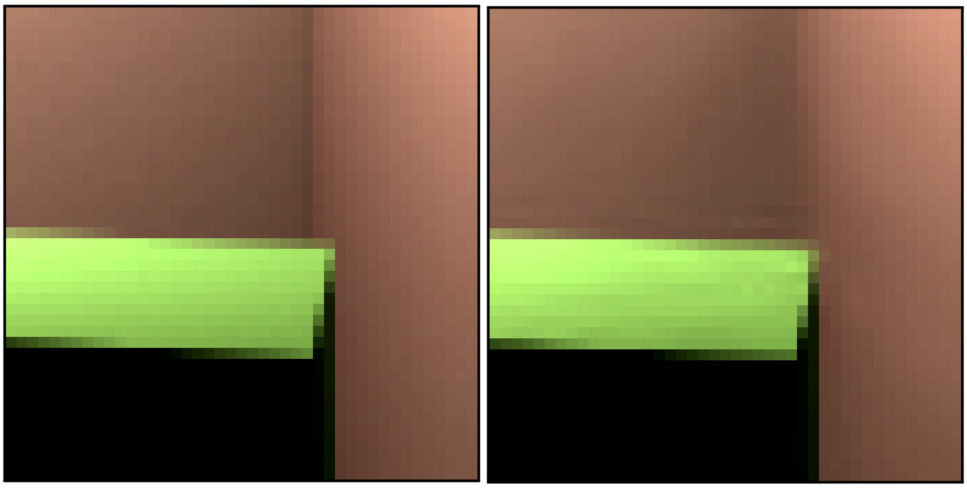


Figure 4.6: Results in `cornell-box` scene. The result of `RHF` is shown on the left while the noisy input is shown on the right. For close-ups on difficult parts see Figures 4.7 and 4.8.

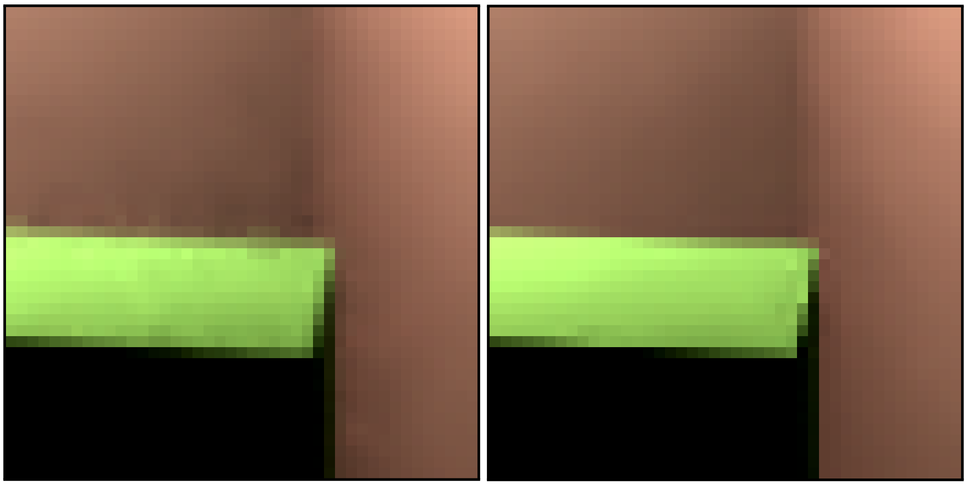


Monte Carlo 256spp 32.7db 14.1s



Reference MC

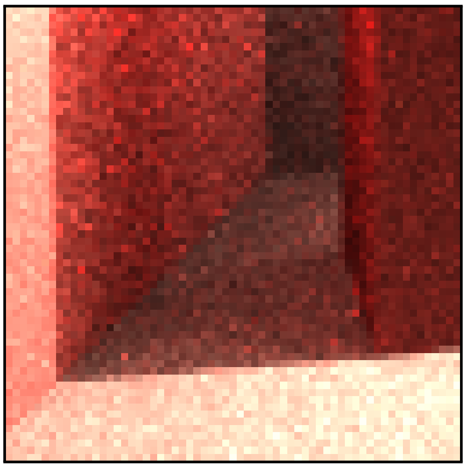
NL-means 44.5db 17.3s (3.2s)



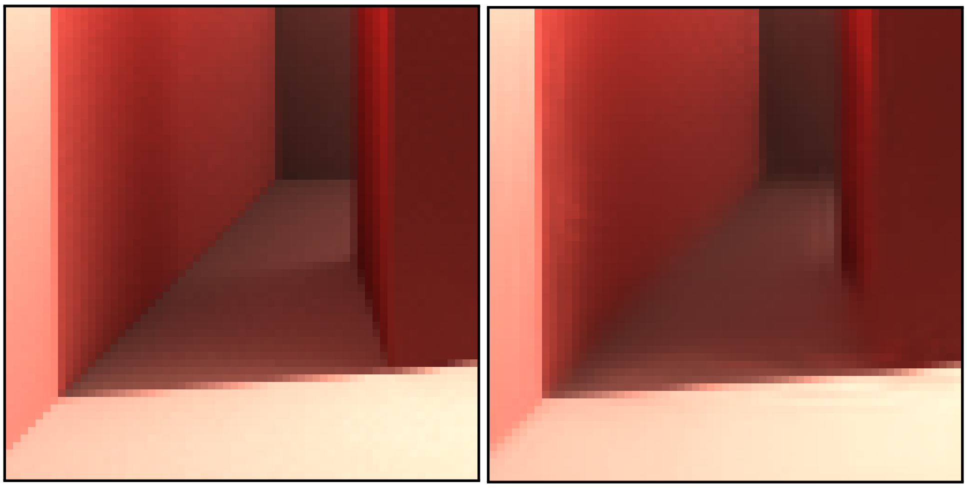
ASR 333spp 42.3db 20.1s

RHF 46.7db 19.6s (5.5s)

Figure 4.7: Results in `cornell-box` scene (close-ups). In all cases, the PSNR values are given for the whole image. In general ASR tends to create artifacts near edges. RHF produces the best PSNR, with no visible artifacts. For NL-means and RHF the indicated time follows the format *total time (filtering time)*.

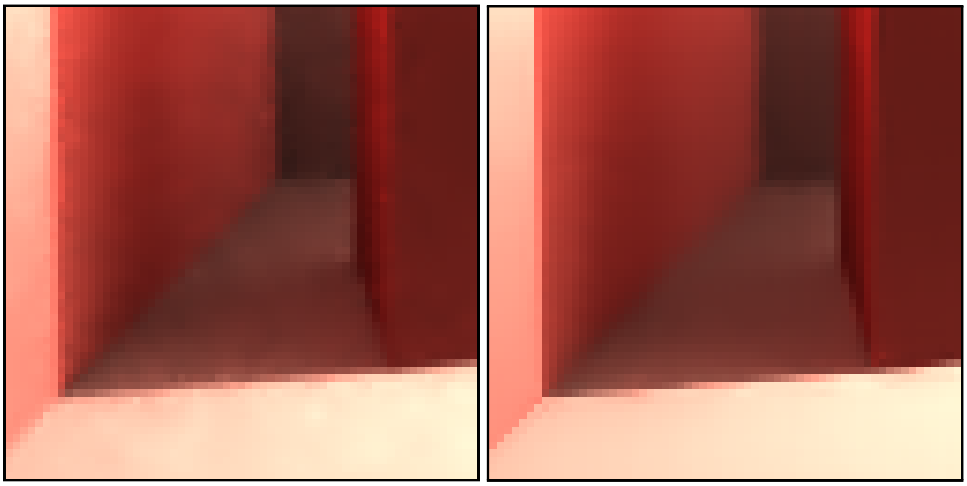


Monte Carlo 256spp 32.7db 14.1s



Reference MC

NL-means 44.5db 17.3s (3.2s)



ASR 333spp 42.3db 20.1s

RHF 46.7db 19.6s (5.5s)

Figure 4.8: Results in `cornell-box` scene (close-ups). In all cases, the PSNR values are given for the whole image. RHF produces the best PSNR, with no visible artifacts. NL-means destroys complex edges such as the one on the bottom left box of the Cornell scene. For NL-means and RHF the indicated time follows the format *total time (filtering time)*.

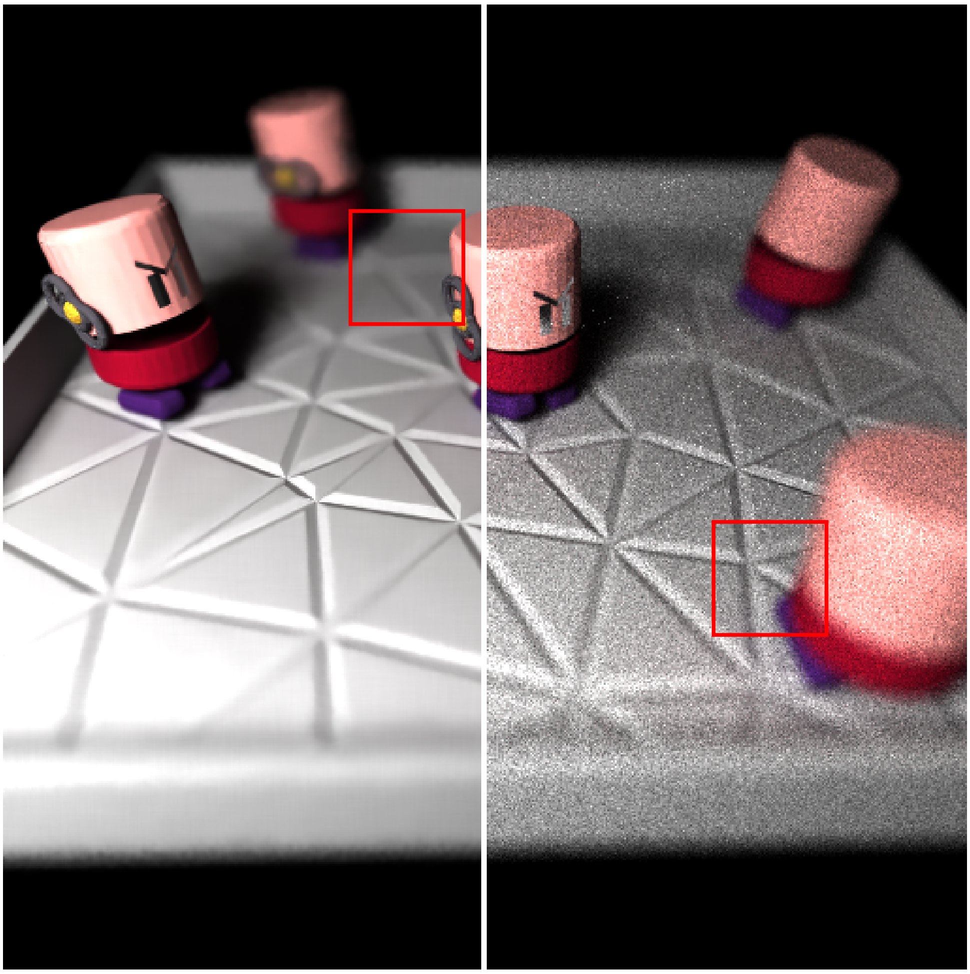
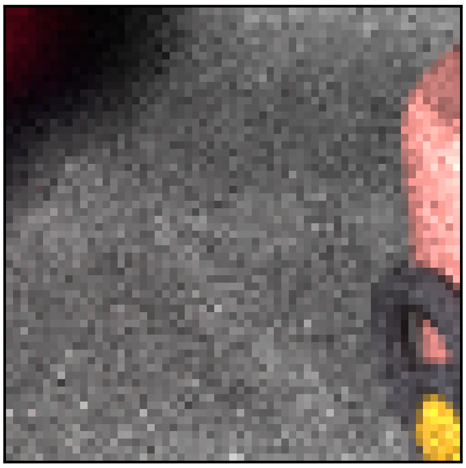
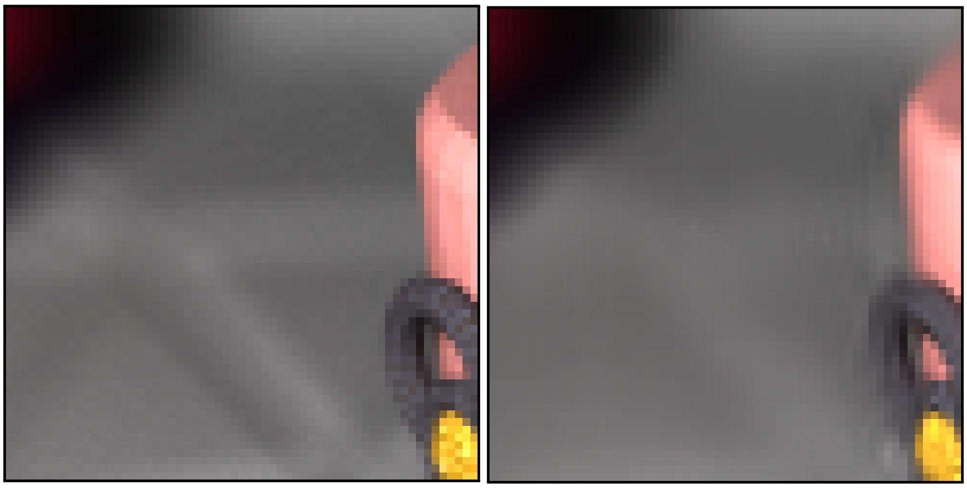


Figure 4.9: Results in `toasters` scene. The result of `RHF` is shown on the left while the noisy input is shown on the right. For close-ups on difficult parts see Figures 4.10 and 4.11.

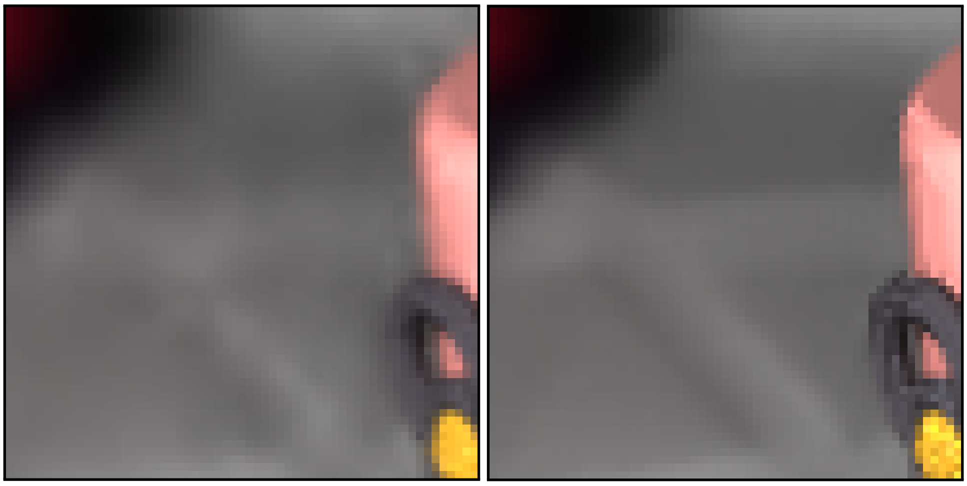


Monte Carlo 256spp 23.4db 64.4s



Reference MC

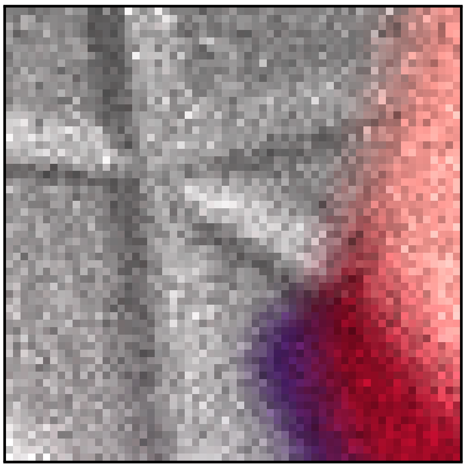
NL-means 35.1db 73.8s (9.4s)



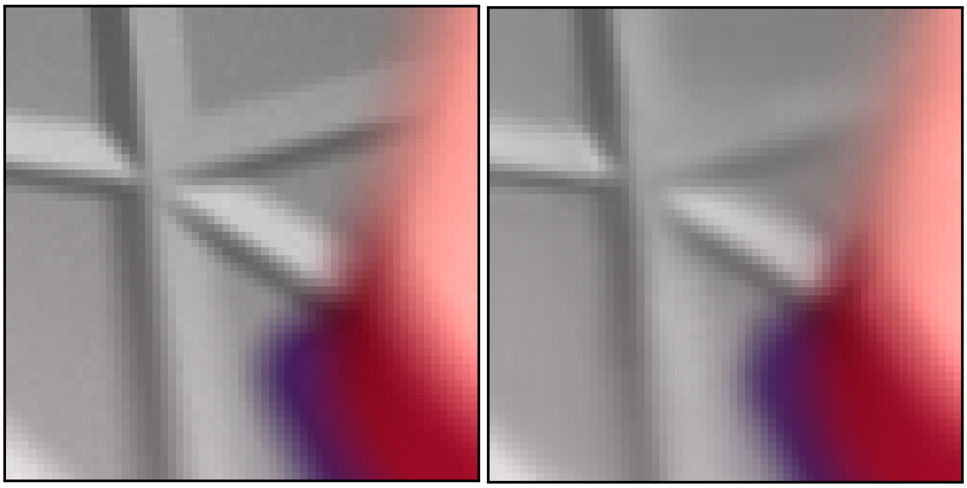
ASR 250spp 35.7db 88.1s

RHF 38.1db 88.8s (24.4s)

Figure 4.10: Results in `toasters` scene (close-ups). In all cases, the PSNR values are given for the whole image. In general ASR tends to create artifacts near edges. RHF produces the best PSNR, with no visible artifacts. For NL-means and RHF the indicated time follows the format *total time (filtering time)*.

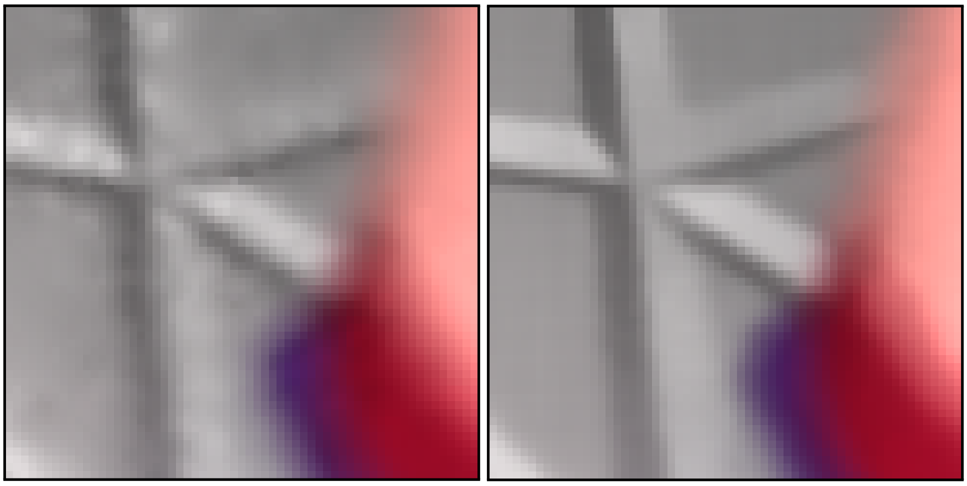


Monte Carlo 256spp 23.4db 64.4s



Reference MC

NL-means 35.1db 73.8s (9.4s)



ASR 250spp 35.7db 88.1s

RHF 38.1db 88.8s (24.4s)

Figure 4.11: Results in `toasters` scene (close-ups). In all cases, the PSNR values are given for the whole image. In general ASR tends to create artifacts near edges. RHF produces the best PSNR, with no visible artifacts. For NL-means and RHF the indicated time follows the format *total time (filtering time).*

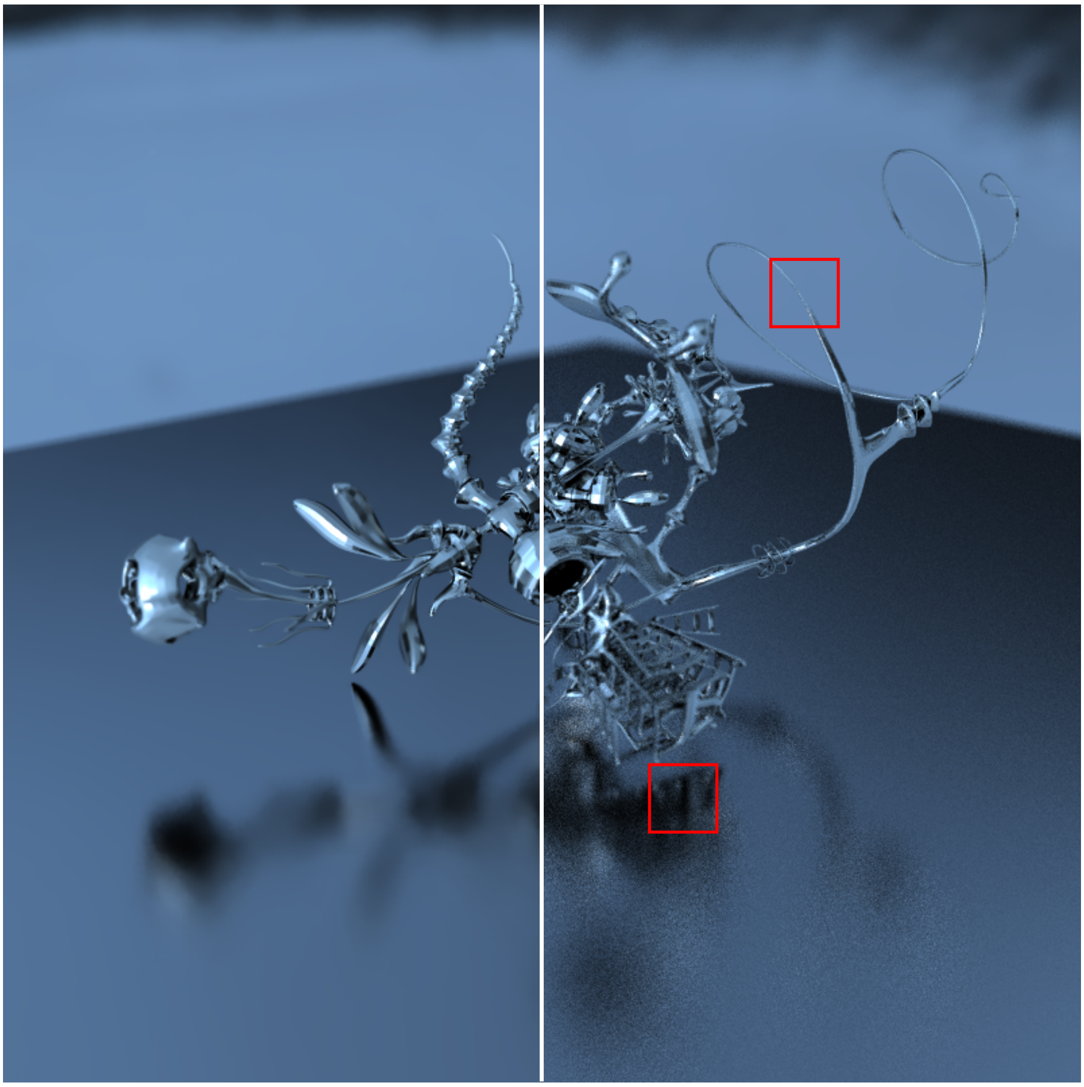
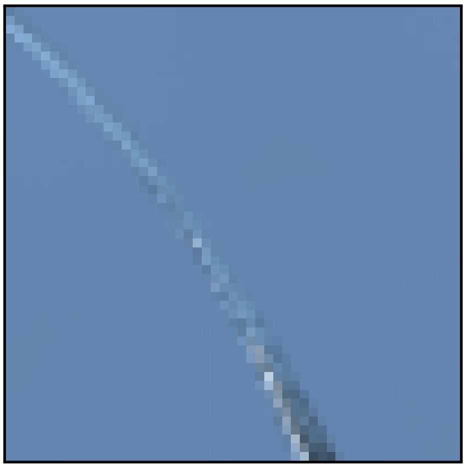
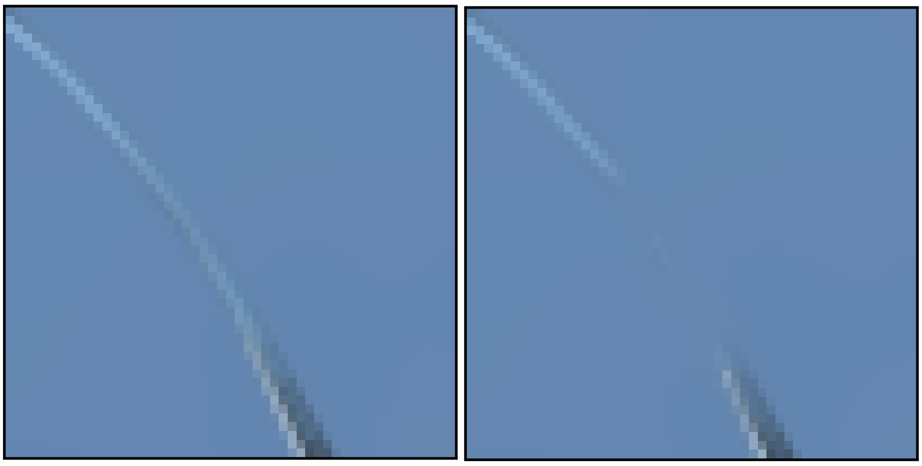


Figure 4.12: Results in `yeahright` scene. Fine geometry details, glossy surfaces and indirect illumination presented in the `yeahright` scene are rendered with the PBRT-V2 path-tracing algorithm. The result of RHF on the left hand part of the image is shown on the left while the noisy input of the right part is shown on the right. For close-ups on difficult parts see Figures 4.13 and 4.14.

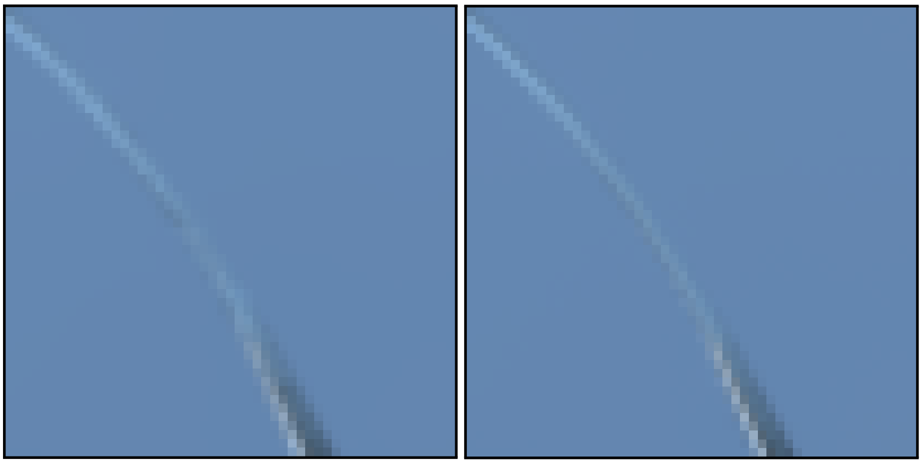


Monte Carlo 64spp 35.4db 39.7s



Reference MC

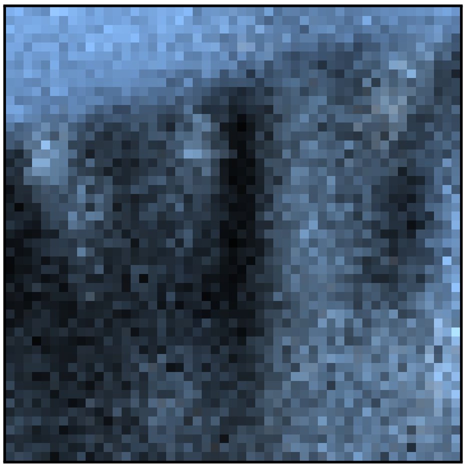
NL-means 41.5db 66.7s (27.0s)



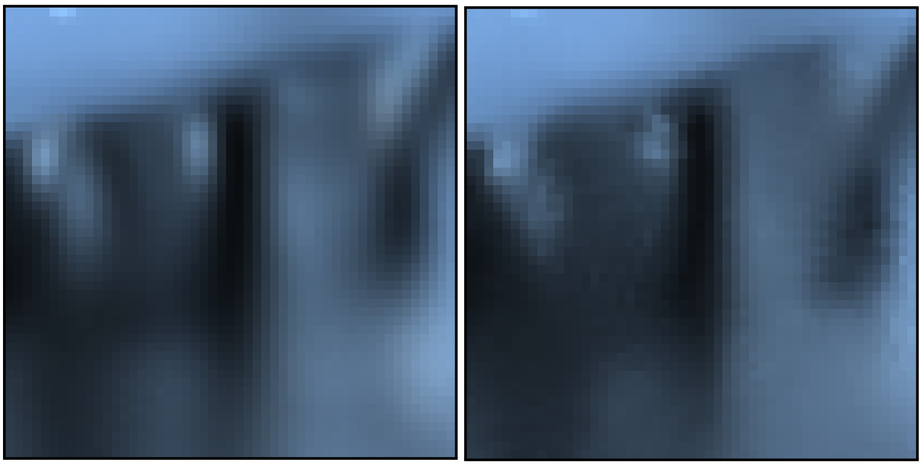
ASR 67spp 42.9db 83.0s

RHF 42.7db 82.6 (42.9s)

Figure 4.13: Results in `yeahright` scene (close-ups). In all cases, the PSNR values are given for the whole image. The comparison is done in such a way that the ASR computational time matches the Monte Carlo samples generation + RHF filtering time. NL-means looses fine structures such as the thin metallic edge. For NL-means and RHF the indicated time follows the format *total time (filtering time).*

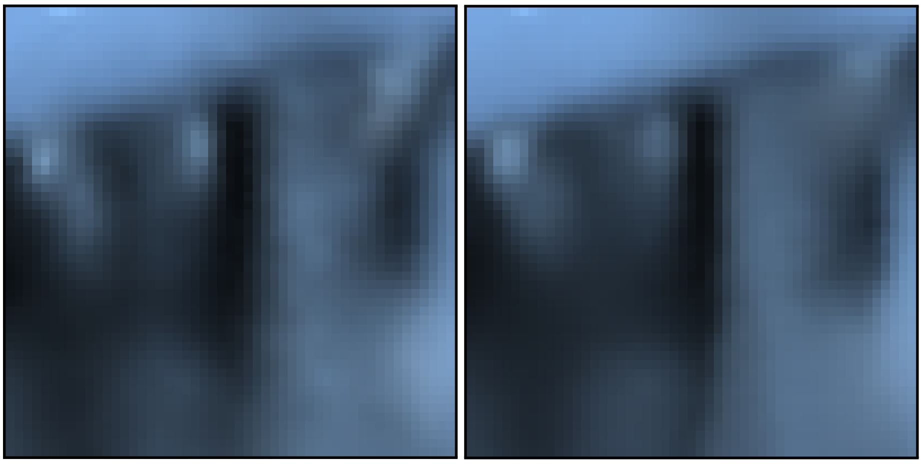


Monte Carlo 64spp 35.4db 39.7s



Reference MC

NL-means 41.5db 66.7s (27.0s)

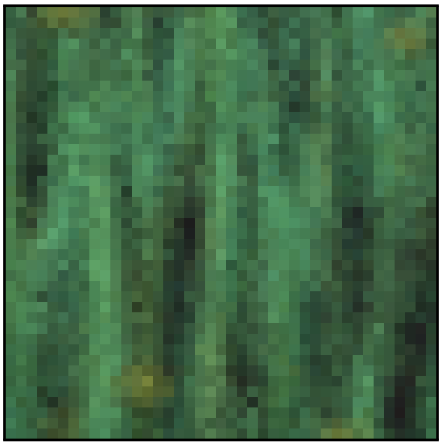


ASR 67spp 42.9db 83.0s

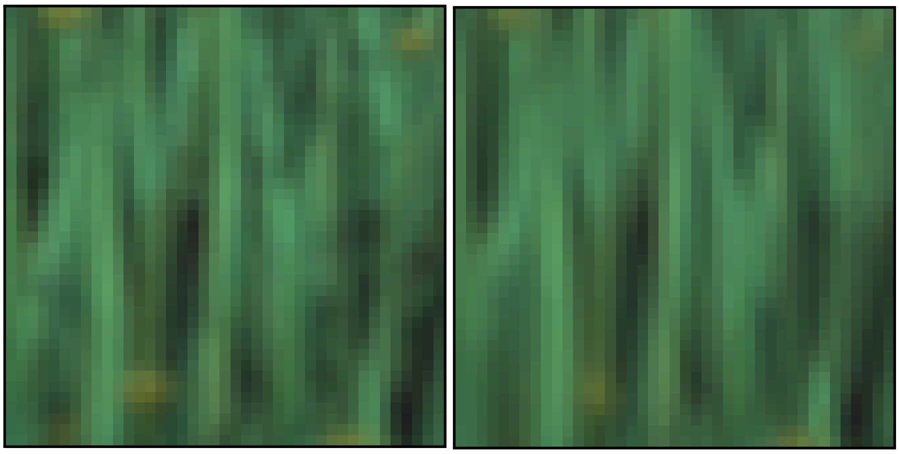
RHF 42.7db 82.6 (42.9s)

Figure 4.14: Results in `yeahright` scene (close-ups). In all cases, the PSNR values are given for the whole image. The comparison is done in such a way that the ASR computational time matches the Monte Carlo samples generation + RHF filtering time. In this particular scene, ASR performs well, but creates artifacts in the shadow. RHF produces a similar PSNR with no artifacts. For NL-means and RHF the indicated time follows the format *total time (filtering time).*

Figure 4.15: Results in `plants-dusk` scene. The result of `RHF` is shown on the left while the noisy input is shown on the right. For close-ups on difficult parts see Figures 4.16 and 4.17.

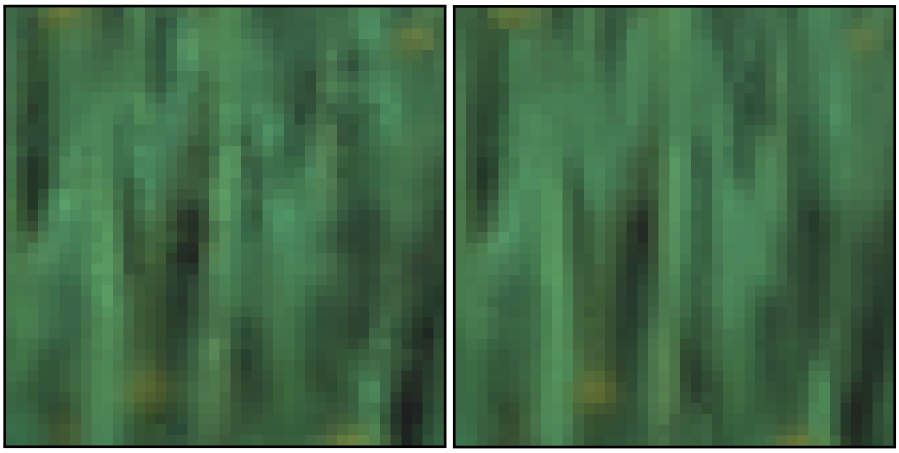


Monte Carlo 64spp 35.4db 39.7s



Reference MC

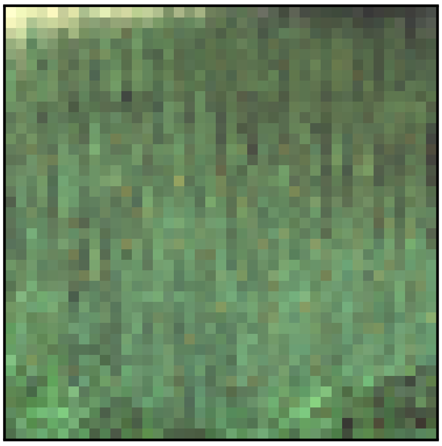
NL-means 41.5db 66.7s (27.0s)



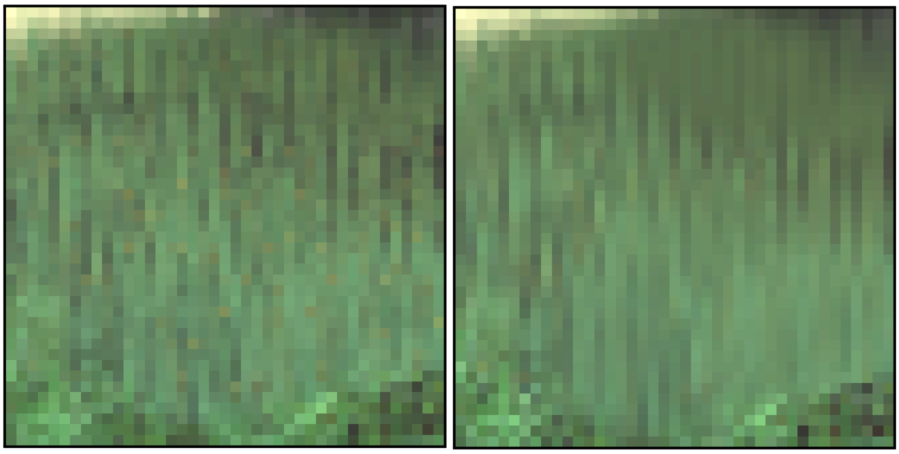
ASR 67spp 42.9db 83.0s

RHF 42.7db 82.6 (42.9s)

Figure 4.16: Results in `plants-dusk` scene (close-ups). In all cases, the PSNR values are given for the whole image. The comparison is done in such a way that the ASR computational time matches the Monte Carlo samples generation + RHF filtering time. This image presents a very complex fine geometry which is very difficult to capture with few samples. Nonetheless, the proposed algorithm produce acceptable quality and PSNR. For NL-means and RHF the indicated time follows the format *total time (filtering time)*.

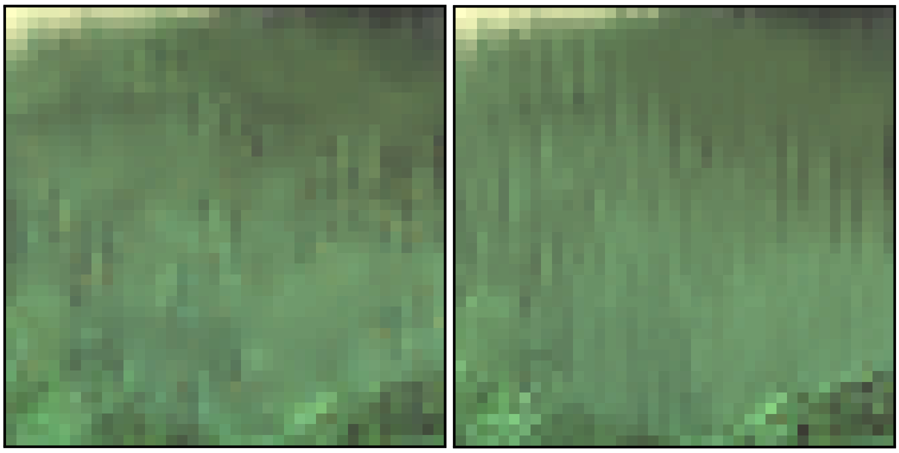


Monte Carlo 64spp 35.4db 39.7s



Reference MC

NL-means 41.5db 66.7s (27.0s)



ASR 67spp 42.9db 83.0s

RHF 42.7db 82.6 (42.9s)

Figure 4.17: Results in `plants-dusk` scene (close-ups). In all cases, the PSNR values are given for the whole image. The comparison is done in such a way that the ASR computational time matches the Monte Carlo samples generation + RHF filtering time. This image presents a very complex fine geometry which is very difficult to capture with few samples. Nonetheless, the proposed algorithm produce acceptable quality and PSNR. For NL-means and RHF the indicated time follows the format *total time (filtering time)*.

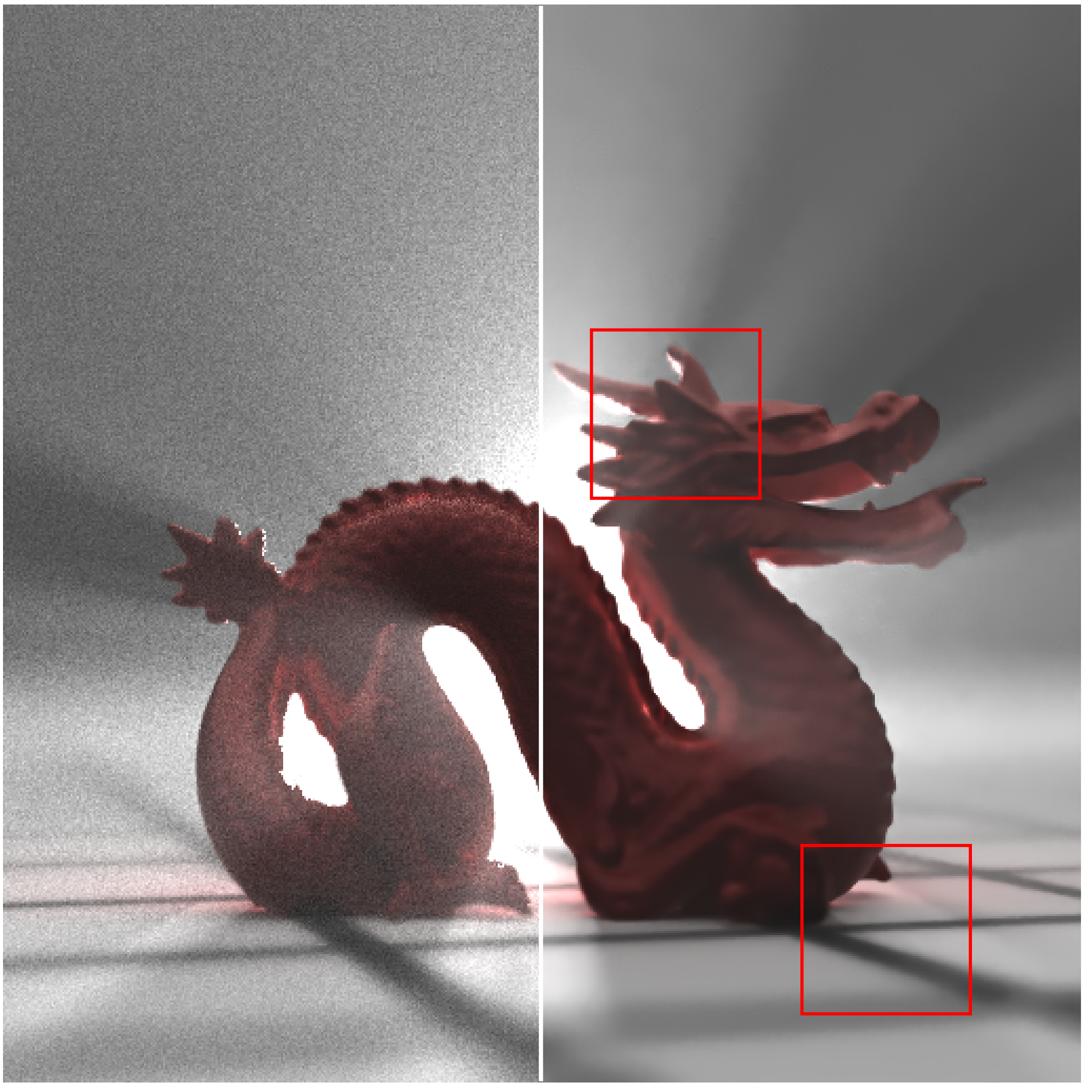
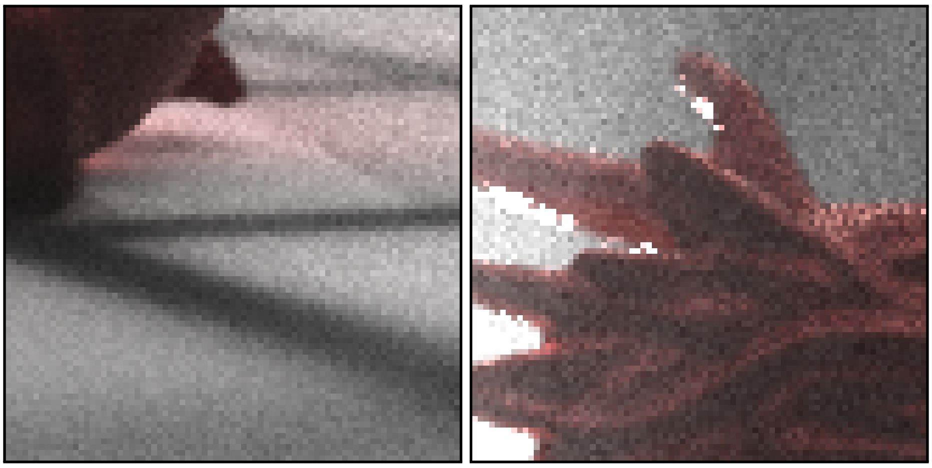
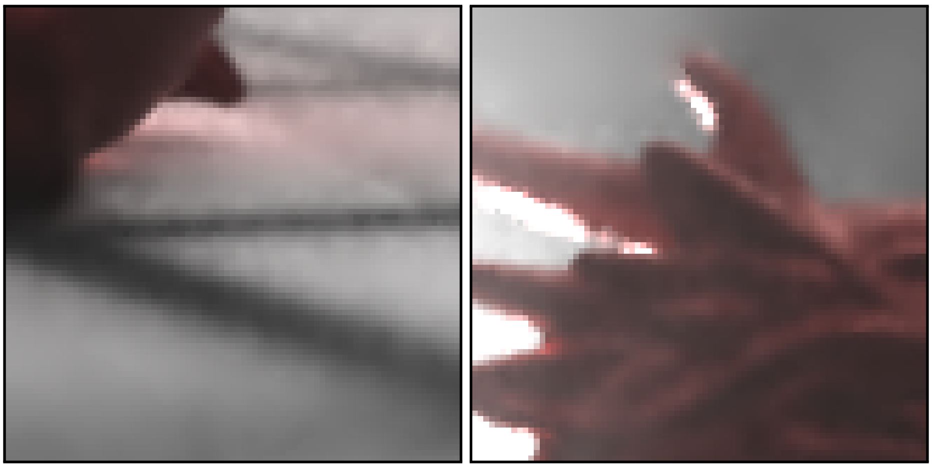


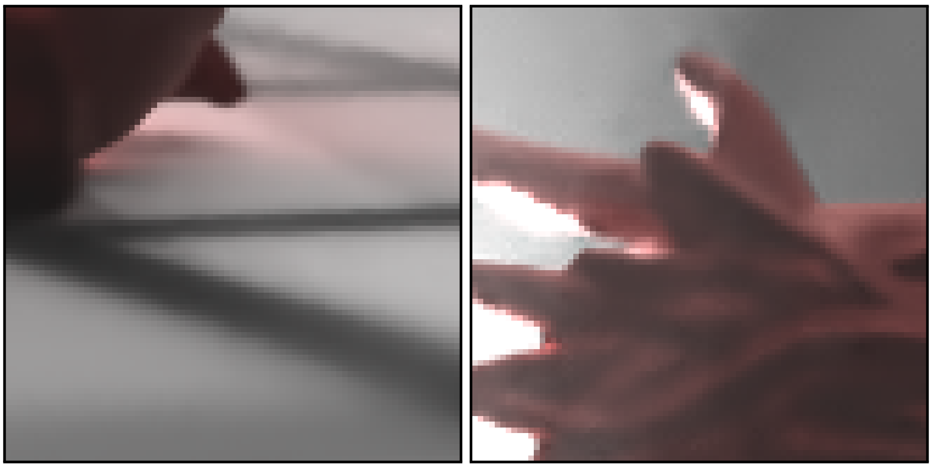
Figure 4.18: Light interaction with participating media rendered through a photon mapping algorithm shows the generality of the proposed filtering. The comparison is done in such a way that the ASR computational time matches the Monte Carlo samples generation + RHF filtering time. For close-ups on difficult part see Figure 4.19.



PM+FG 378s



ASR 414s



RHF 408s

Figure 4.19: Results in `dragon-fog` scene (close-ups). The comparison is done in such a way that the ASR computational time matches the Monte Carlo samples generation + RHF filtering time. Since Photon Mapping is a biased rendering algorithm no reference image (and therefore no PSNR) was computed.

4.6 Discussion, Limitations and Future Work

The maximum distance authorized between two patches plays an important role in the bias-variance tradeoff of the method. If the threshold is set in a conservative way, then very few pixels will be averaged. Thus, the filtering stage will not introduce bias, but the variance reduction will be low. On the other hand, if set too large many pixels of different nature would be considered similar, and averaged by error. Then the resulting image would be smooth but also biased (see Figure 4.20).

If we accept that the selection of the most similar pixels for each noisy input pixel is independent of the number of samples, then the gain in PSNR when casting more samples is only due to the averaging of less noisy pixels. By the randomness of the MC rendering, the noise of the input pixels is reduced by +3db/octave, thus the ideal (best) slope should be +3db/octave. This is the ideal, because it assumes that there is no error in the selection of similar pixels. Therefore, we can consider the difference in slope to the ideal +3db/octave as a measure of *experimental bias* (introduced error). While the proposed algorithm RHF has an experimental bias of 0.2db, the NL-means bias is three times larger. More important, the relative bias to the MC gain is $0.2/3 \approx 0.07$, which demonstrates that the proposed algorithm makes very few wrong ray color attributions.

Moreover, the proposed algorithm is consistent up to the discretization of the color distribution. As the number of samples increases, more evidence is required to average two pixels. In the limit two pixels will be averaged only if their color histograms are the same. Therefore, in practice, as the number of samples grows the method converges to the expected solution, as illustrated by the experiment in Figure 4.21.

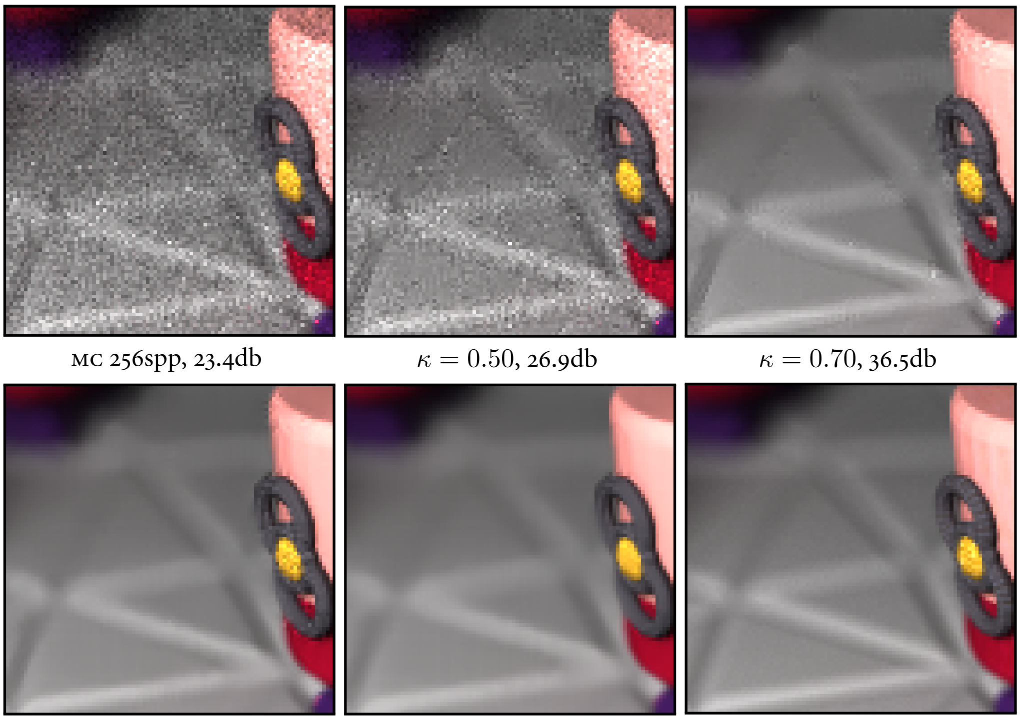
The acceleration factor depends on the degree of self-similarity of the scene, which fortunately is usually high [Lebrun et al. 2012]. Besides, in order to capture details, pixels need a large enough number of color samples to be well characterized. This is actually a design decision: we wanted our method to produce unbiased high quality images for any kind of scenes and complex effects, and this naturally requires a proper sampling of the light field. If this requirement is not met, the algorithm may not properly cluster similar pixels and details may be removed due to over-blur, as it happens with some details in the `plants` image. In the case of very low sample numbers, if the path-space is regular enough to be well described by sparse sampling, methods based on strong scene hypothesis that use fat samples [Sen and Darabi 2012; Lehtinen et al. 2012] are certainly much more adapted.

4.7 Conclusion

In this work we have introduced RHF, an adaptive filtering scheme that accelerates Monte Carlo renderers. In the proposed approach, each pixel in the image is characterized by the collection of rays that reach its surface. The proposed filter uses a distance based on the sample color distribution of each pixel, to decide whether two pixels can share their samples. This permits to boost the performance of a Monte-Carlo render by reusing samples without introducing significant bias.

We have presented several experiments showing that RHF achieves artifact-free high quality noise reduction on a variety of scenes, and is able to cope with multiple simultaneous effects. The method is not only capable of removing high frequency noise: thanks to its natural multi-scale design, it can also successfully remove low-frequency noise. The proposed method can be easily extended to process animated sequences.

The method is independent of the rendering system and can be applied to samples generated by different methods, such as pure Monte Carlo path-tracing or photon-mapping with final gathering. It could also be potentially applied to post-process other methods that re-synthesize samples



MC 256spp, 23.4db

$\kappa = 0.50$, 26.9db

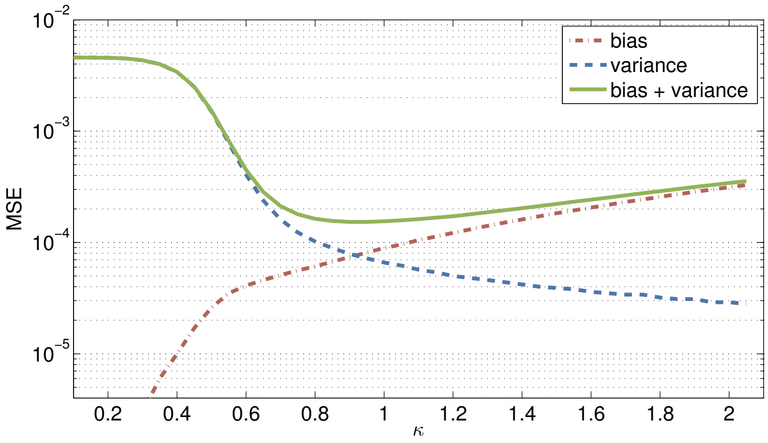
$\kappa = 0.70$, 36.5db

$\kappa = 0.90$, 38.1db

$\kappa = 1.30$, 37.2 db

Reference MC 65536spp

(a)

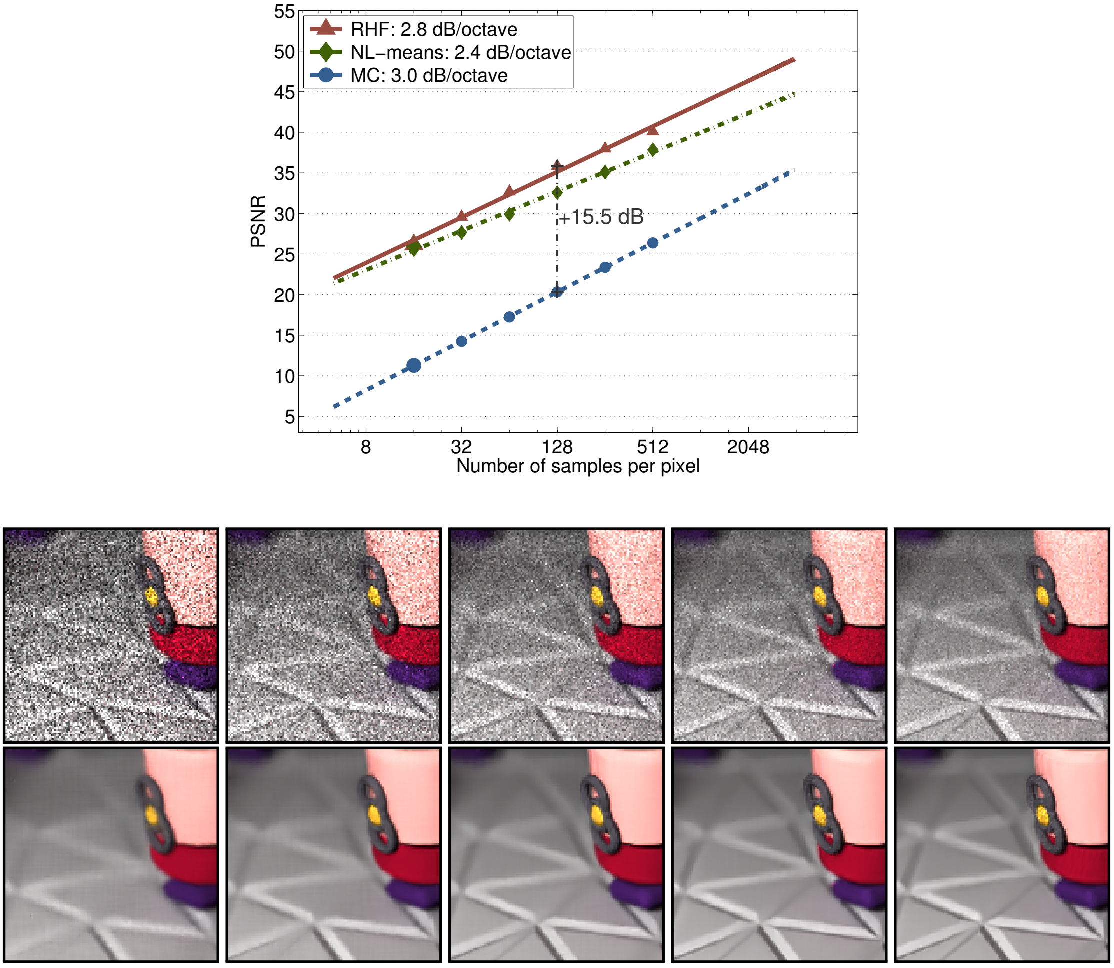


(b)

Figure 4.20: Changing the distance threshold κ: it fixes the maximum distance that two color distributions can differ. (a): a small detail in the `toasters` image filtered with the RHF algorithm with growing κ values. The MSE presents a minimum for $\kappa = 0.7 - 1.0$ (b): If κ is too small the test on the similarity is excessively conservative, and the noise is not reduced (high variance). If κ is large, too many pixels are averaged and the image is blurred (high bias). The results were calculated on the `toasters` scene generated with 256 spp.

using information from the scene, like the one recently proposed by Lehtinen et al. [2012]. An advantage of the proposed filter is that its time and memory complexities do not depend on the number of input samples, and scale linearly with the image size.

Finally, since a direct output of our method is the number of similar pixels for each given pixel, a decision on where to distribute new samples can be adopted. This may lead to an adaptive rendering version of the proposed filtering approach.



32spp 64spp 128spp 256spp 512spp

Figure 4.21: Top: A comparison of the PSNR for RHF, NL-means, and the pure MC pathtracing algorithm on the `toasters` scene. As the number of samples per pixel increases the PSNR increases. In a pure MC scenario the square error decreases linearly with the number of samples (which is a trivial consequence of averaging independent samples). Thus, duplicating the number of samples produces a 3db gain. Although for the proposed algorithm the slope gain is a little smaller (2.8db/octave), RHF reduces the error significantly in comparison to MC. +15.5dB is a huge difference; it permits to reach the same image quality with $35\times$ fewer samples. On the bottom we show a close up, generated with a varying number of samples through pure MC (top) and filtered with RHF (bottom).

Conclusions and Perspectives

This dissertation concerns the digital image formation process and the quest for

(i) mathematical models to better describe the kind of images that are obtained by this process, and

(ii) algorithms to improve image quality based on such models and to estimate the parameters of those models.

More precisely, we focus on the problems of denoising and blur estimation and address them for images that have been generated via two fundamentally different formation processes. In part I we deal with optically generated images, that have been produced by real photons after conversion to digital information by the sensors in a real camera. In contrast, part II is concerned with images that have been synthetically generated by computer simulations of virtual light hitting a digital 3D model and captured by a virtual camera.

We have introduced a mathematical model of an optical digital camera that provides a framework to evaluate the quality of PSF estimation patterns. The key idea is to find the pattern that makes the PSF estimation inverse problem as well-posed as possible. Or in other words, which is the pattern that carries the most information for the PSF estimation problem. Indeed, the theoretical analysis concludes that a random pattern made from Bernoulli noise is nearly optimal.

An algorithm for estimating the PSF from a photograph of this pattern is proposed. The method is very strict in the precautions to compensate for variations in illumination and possible geometrical distortions between the ideal pattern and the observed image. The procedure is successful, but its setup is somehow tedious since a calibration pattern must be printed and properly assembled. Very accurate estimates of the PSF are achieved, as it is amply shown in the various comparative experiments (the relative error is of the order of 2% to 5%). As usual, a locality-accuracy trade-off had to be resolved. Locality of an order of a few hundred pixels can be achieved under common noise conditions.

Within the same mathematical framework, we confirmed that traditional state-of-the-art slant-edge based methods lead to inverse problems that are ill-posed and require some kind of regularization. This is a major inconvenient, since introducing regularization considerably biases the estimation.

To avoid the use of a pattern, we introduced a *semi-blind* algorithm for the subpixel estimation of the camera PSF that uses 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 blur between them.

The PSF estimation method is regularization-free, being closely related to the Bernoulli pattern algorithm. Nevertheless, the question is far more intricate since only the relative blur between the acquisitions can be estimated. The algorithm is based on a mathematical analysis that proves that the PSF can be recovered from the relative blur.

Images of planar textured scenes with a flat enough spectrum are necessary to reach high accuracy. Fortunately, as was experimentally shown, many textures found in nature are well adapted to these requirements. This semi-blind algorithm reaches similar accuracy levels to the one with the near-optimal Bernoulli pattern, with the advantage of not requiring any special acquisition setup or calibration pattern, thus being much more practical.

The experiments here suggest that for typical DSLR cameras, each color channel is undersampled with respect to the ideal Nyquist rate given by the PSF, by a factor of 2 or even 4. This fact was confirmed, even with DSLR models including an optical anti-aliasing filter on the sensor. This more than justifies *a posteriori* the need of a subpixel estimation procedure.

Accurate PSF estimates are important for several classical image processing applications, such as image superresolution, image de-blurring, and camera quality evaluation. In what follows we present some insights of possible future research directions regarding these and other applications.

In the current conditions, the camera PSF acts as a strong low-pass filter attenuating significantly all frequencies beyond two times the camera's channel sampling frequency. From the point of view of the Shannon-Nyquist theory, this sets an upper limit in the gain of multi-image superresolution that is not too attractive.

In a modern DSLR camera with no significant lens aberration, working at a wide aperture (e.g., f-number ≤ 5.6), the PSF is mainly due to the optical anti-aliasing filter and the light integration in the sensor array. Hence, it becomes fundamental to remove this optical filter, that has been introduced to mitigate the aliasing artifacts, to unveil the latent superresolution potential that a digital camera may have. Some companies in the market, such as max-max[1], offer to remove the camera optical anti-aliasing filter and to replace it by an ordinary glass with the same refractive index. Recently, Nikon has introduced a new model (D800E) of its D800, which is essentially the same but does not apply the optical anti-aliasing filtering. Although not a majority, some camera manufacturers are already considering the possibility of doing in-camera superresolution. Hasselblad developed a camera (H4D-200MS) capable of producing images of $4\times$ the sensor resolution by fusing several snapshots with the sensor subpixically shifted.

The problem of image superresolution and deconvolution with a spatially variable PSF are intertwined. Traonmilin et al. [2012] analyzed the problem of superresolution, and showed as an example how to make independent color channel demosaicking under the simplistic hypothesis that the deconvolution and superresolution are decoupled. However, as the authors claim, this is only accurate if the PSF is space invariant and the the snapshots are purely translational motions. Having a tool to estimate locally and accurately the PSF can unlock this problem and allow a multi-image camera to be used as a true measuring instrument without aliasing.

Plenoptic cameras, which sample the 4D light field, have proved to be useful for single image digital refocusing or depth estimation (e.g., the Lytro camera[2]). However, this richer sampling comes at a significant associated cost in spatial resolution. Some researchers [Bishop et al. 2009; Bouman et al. 2011; Bishop and Favaro 2012]) have recently explored to extend the resolution of these cameras combining all the information captured from the light field. A mathematical modeling of the acquisition process, along with a local estimate of the PSF is essential to achieve this goal.

Another related problem is the design of fast autofocus algorithms. It then becomes essential to make a very fast blur measure, especially in the case of autofocus in video, and to decide in which

---

[1] www.maxmax.com

[2] www.lytro.com

direction and how far the focal point should be moved to capture an in-focus image. In addition, a precise calibration of the out-of-focus PSF could be very useful for the problem of estimating the depth from defocus. Currently, the in-focus kernels are oversimplistically modeled as Gaussian or circular kernels. However, it is well known by experimented photographers that these kernels, which give rise to the popularly known *bokeh* effect, have much more complex shapes. Coded-aperture and coded-exposure imaging (see e.g., [Raskar et al. 2006; Levin et al. 2007]) allow the point spread function to be engineered for better conditionate specific applications such as motion deblurring or depth-from-defocus. Maximizing the quality of the restored image requires an accurate estimation of the resulting PSF. We would like to explore and quantify the impact of the PSF estimation in these problems.

Another useful application and a possible future research direction of having accurate estimate of the PSF is to know the types of aberrations that an optical camera introduces. The proposed accurate non-parametric PSF estimate can be matched to parametrical simulations of different types of aberrations to conclude which type of aberrations the camera presents (such as, comma, astigmatism, chromatic blur). This information can be helpful to quantitatively assess the quality of a device, but also to better design the optical system.

Detecting the parts of an image that were poorly sampled (leading to aliasing) and restore them is an important problem in image processing. The work of Coulange and Moisan [2010] proposed to detect, by analyzing the Fourier spectrum, those harmonics that are likely to be aliased. One possible avenue of research is trying to use the this technique along with the work of Almansa et al. [2004] for image restoration with samples in non-rectangular lattices to make single-image antialiasing for traditional and non-traditional light sensors.

Last, but not least, online PSF estimation is of significant importance in the calibration of modern astronomical telescopes. Most of these telescopes have a large mirror (of several meters of diameter) which is exposed to variable stress due to unpredicted weather conditions. The only way to build and maintain in service such a large system is to work in an active optics principle: the mirror has actuators that modify its shape (and thus the wave front and the PSF) proactively to prevent deformations. Image analysis and continuous PSF re-estimation becomes necessary to adapt the optical shape and obtain a PSF as sharp as possible.

The second part of this thesis addresses a denoising problem in image synthesis. Although there have been several breakthroughs very recently, synthesizing high quality realistic images in a reasonable amount of time remains a major challenge in computer graphics. In this dissertation, we introduced an adaptive filtering algorithm for accelerating Monte Carlo renderers. In the proposed approach, each image pixel is characterized by the set of rays that reach its surface. The algorithm, that we named *ray histogram fusion* (RHF), uses a similarity measure on the empirical ray color distribution of each pixel, to decide whether two pixels can be fused.

This simple procedure permits to boost the performance of any stochastic renderer by reusing samples without introducing significant bias. The proposed method achieves artifact-free high quality noise reduction on a variety of scenes, and is able to cope with multiple simultaneous rendering effects. The method is renderer independent and can be applied to samples generated by different methods, such as pure Monte Carlo path-tracing or photon-mapping with final gathering.

Thanks to its natural multi-scale design, it can successfully remove noise at all scales. Moreover, the algorithm is easily extended to process videos from animated sequences. One major advantage of the ray histogram fusion filter is that its time and memory complexities are independent of the number of input samples, and it scales linearly with the image size.

As a future work, we would like to investigate how RHF can be applied to post-process other methods that re-synthesize samples using information from the scene, like the one recently proposed by Lehtinen et al. [2012]. Also, since a direct output of the method is the number of similar pixels that each pixel has, a decision on where to distribute new samples can be adopted. This may be the basis of an adaptive rendering version of the proposed filtering.

The proposed RHF filter assumes that the pixels grouped as similar have exactly the same expected value. In practice, this does not strictly hold and can therefore lead to the introduction of a small bias. We would like to explore whether the use of more general models (e.g., affine or more complex statistical models) can improve the performance and keep bias controlled.

Besides, it would also be interesting to explore different ways of reducing the computational cost of the RHF filter. Recently, Gastal and Oliveira [2012] have introduced a technique for accelerating filters based on the auto-similarity principle. Their algorithm, which reaches outstanding results, learns a set of manifolds that well capture the image structure, and then filters each of them separately. Hence, this is a natural research direction to reduce the computational cost of the RHF filter.

A Detailed Description of the PSF Estimation Algorithms

This section presents a complete detailed algorithmic description of the proposed non-blind and two-scale photographs PSF estimation procedures. A demo facility and a reference source code can be found in the respective IPOL publications [Delbracio et al. 2012a,b].

A.1 PSF Estimation from a Calibration Pattern Image

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.

In the next paragraphs we present a brief summary for each block.

Pattern detection

In order to detect the pattern in the image we use the line segment detector (LSD) algorithm by Grompone von Gioi et al. [2012]. The idea is to detect the segments that are present in the pattern structure as can be seen in Figure A.1.

By using the detected line segments (in Figure A.1 are shown in red, green and blue for illustration purposes) 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:

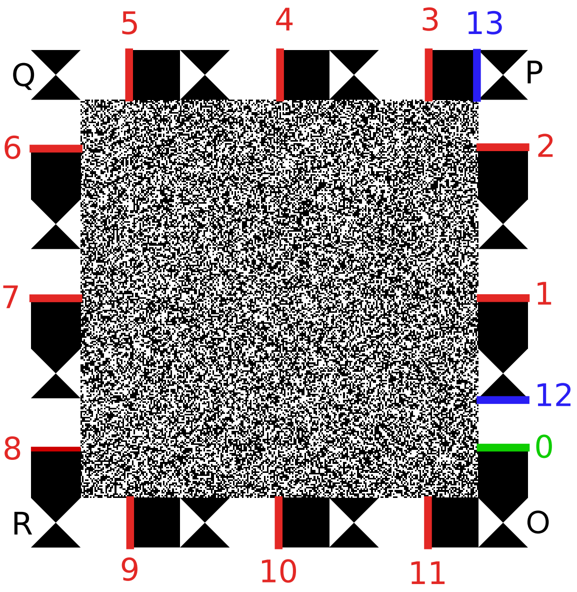


Figure A.1: The Bernoulli pattern and the line segments used for the detection of the pattern in an input image (shown in red, green and blue).

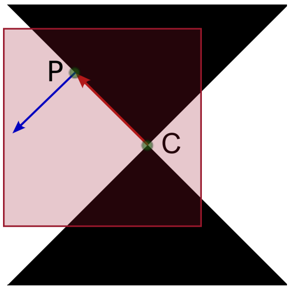
   i. Compute the distance between s_{ij} and the ideal relative position of every searched segments (red and blue segments).
   ii. If it is up to a tolerance and the distance is less than in the previous detection of the same segment, update the segment and the distance to the optimal position

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.

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 [Bradski 2000]. 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

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

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 ∇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 (A.1).
5. If $\|C - C_{\text{old}}\| <$ tol or ++iter $>$ max_iter exit, else go to 1.

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

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 [Sprengel et al. 1996] to model the whole deformation. Since we have previously detected the $\{\tilde{p}_i\}$ checkerboard corners centers from the observed pattern 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:

$$E = \sum_i \|f(P_i) - \tilde{p}_i\|^2 + \lambda \iint \left(f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2\right) dxdy,$$

where λ controls the amount of regularization. As we will show if λ 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 Φ the $n \times n$ matrix formed from $\Phi(\|P_i - \tilde{p}_j\|), \quad i, j = 1..n$, representing the thin-plate kernel.

It can be shown (see the work by Sprengel et al. [1996]) 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,

$$X = \left[\begin{array}{cc} Q_1 & Q_2 \end{array}\right] \left[\begin{array}{c} R \\ 0 \end{array}\right].$$

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 100×100 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.

Illumination estimation and normalization

In order to recover the PSF, 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:

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

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 $u \to \alpha u^2 + (1-\alpha)u$ where α is chosen so that the mean of the pattern after the correction is 0.5. Hence, α can be calculated as:

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

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

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 4×4 block of pixels. We are going to work directly with these samples **u** (digital image), instead of the continuous pattern.

2. Frequencies higher than $s\pi$ are cut off from the digital pattern **u** to get **ũ**. The DCT of **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 **ũ** by applying the Inverse DCT. Note that π 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 **ũ** is scaled to the desired resolution s by bicubic interpolation. The resultant image $\tilde{\mathbf{u}}_D$ is of size $(s \times (m-1)+1, s \times (n-1)+1)$.

Building the linear system

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

$$\underset{\mathbf{h}}{\arg\min} \quad ||\mathbf{S}_s \mathbf{C}[\tilde{\mathbf{u}}_D]\mathbf{h} - \mathbf{v}||_2^2. \tag{P}$$

The matrix $\mathbf{S}_s \mathbf{C}[\tilde{\mathbf{u}}_D]$ is composed by:

1. The matrix $\mathbf{C}[\tilde{\mathbf{u}}_D]$ 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.

Numerical methods for PSF estimation

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

$$\underset{\mathbf{h}_i}{\arg\min} \quad \|\mathbf{S}_s\mathbf{C}[\tilde{\mathbf{u}}_D]\mathbf{h}-\mathbf{v}\|_2^2 \quad \text{subject to} \quad \mathbf{h}_i \geq 0,\ i=1,\dots,r^2. \tag{P}$$

To directly solve the non-negative least squares problem (P) we used the Newton interior point algorithm proposed by Portugal et al. [1994] 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 reference source code can operate in any of these three options.

A.2 PSF estimation from Two Photographs at Different Distances

The *input* of the algorithm are the two digital images: $\tilde{\mathbf{v}}_1, \tilde{\mathbf{v}}_2$, the superresolution factor s and the kernels (inter-image kernel and PSF) support size: $p \times p$ at the $s\times$ superresolved grid. The *output* of the algorithm are a $s\times$ sampling of the inter-image kernel **k** and a $s\times$ sampling of the camera PSF **h**. Both images are of size $p \times p$.

Image subpixel alignment and 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 by Moisan et al. [2012]. These subroutines may be replaced by any other accurate subpixel 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 $m' \times n'$ and $m \times n$. Then

$$D : [0, m'-1] \times [0, n'-1] \to [0, m-1] \times [0, n-1].$$

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

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

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

In order to generate the rescaled samples $\mathbf{H}_{\lambda/s}\tilde{\mathbf{v}}_1$ from the digital image $\tilde{\mathbf{v}}_1$ we need to interpolate it at the desired scale λ/s . This is done by using the estimated geometric transformation D.

From now on we will consider that $1\times$ 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 zoom-in image $\tilde{\mathbf{v}}_1$. The DCT of $\tilde{\mathbf{v}}_1$ 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 common region part in the captured zoom-out image). Then we recover the filtered version by applying the inverse DCT. Note that 2π represents the camera sensor sampling frequency that is why the factors m and n are included.

2. Using the previously computed geometric transformation D, the filtered zoom-in image is bicubically interpolated at the desired resolution λ/s. We do this by estimating the image values at a regular $s\times$ grid: $[0, m'-1] \times [0, n'-1]$ (i.e., the step size is 1/s). The resultant image $\mathbf{H}_{\lambda/s}\tilde{\mathbf{v}}_1$ is of size $(s \times (m-1)+1, s \times (n-1)+1)$.

Solving for the inter-image kernel

First, the following linear system is built:

$$\arg\min_{\mathbf{k}} \quad \|\mathbf{M}\mathbf{S}_s\mathbf{C}[\mathbf{H}_{\lambda/s}\tilde{\mathbf{v}}_1]\mathbf{k} - \mathbf{M}\tilde{\mathbf{v}}_2\|_2^2, \tag{Pi}$$

where the matrix $\mathbf{M}\mathbf{S}_s\mathbf{C}[\mathbf{H}_{\lambda/s}\tilde{\mathbf{v}}_1]$ is composed by:

1. The matrix $\mathbf{C}[\mathbf{H}_{\lambda/s}\tilde{\mathbf{v}}_1]$ 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 $p \times p$.

2. The s-down-sampling matrix $\mathbf{S}_s$ takes one sample per each block of $s \times s$ pixels.

3. Finally a mask $\mathbf{M}$ is applied that puts to zero all values that are outside the *region of interest*.

The *region of interest* consists of a trapezoidal mask that restricts the convolution to the common part in the two images. Also this mask is eroded by a factor $\frac{p-1}{2s}$ to avoid boundary problems due to the convolution of finite support sequences.

We rewrite the zoom-out observed image as a vector to be consistent with the matrix formulation of the system. The mask $\mathbf{M}$ restricting the image to the region of interest is also applied to the observation $\tilde{\mathbf{v}}_2$. Next, we need to solve Problem (Pi) which is simply a least squares algorithm.

From the inter-image kernel to the the PSF

In order to recover the camera PSF h we need to compute (see Section 3.4)

$$\mathbf{h} = \lim_{n\to\infty} \mathbf{H}_{\lambda^n}(\mathbf{k} * \mathbf{H}_{\frac{1}{\lambda}}\mathbf{k} * \cdots * \mathbf{H}_{\frac{1}{\lambda^n}}\mathbf{k}).$$

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 to $\lambda = (h_{0,0}, h_{1,1})$. 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 $\lambda = (\lambda_x, \lambda_y)$ (bicubic interpolation).
3. Calculate $\mathbf{u}_n = \mathbf{H}_{1/\lambda^n}\mathbf{k} * \mathbf{u}_{n-1}$.
4. If $\min\{\lambda_x^n, \lambda_y^n\} > \lambda_{\max}$ go to 5. Else update $n := n + 1$ and repeat from 2.
5. Calculate $\mathbf{h} = \mathbf{H}_{\lambda^n}\mathbf{u}_n$ (bicubic interpolation).

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

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.

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 superresolution factor is always less than the relative distance, so for $4\times$ estimation relative distance should be higher than 4.
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”[1]. A single color channel should then be extracted and use as the input to our algorithm.

---

[1] DCRAW is the Dave Coffin’s utility for decoding raw digital photos in Linux.

Bibliography

Adams, A., Gelfand, N., Dolson, J., and Levoy, M. Gaussian kd-trees for fast high-dimensional filtering. *ACM Trans. Graph.*, 28(3):21:1–21:12, July 2009. ISSN 0730-0301. doi: 10.1145/1531326.1531327.

Almansa, A., Durand, S., and Rougé, B. Measuring and Improving Image Resolution by Adaptation of the Reciprocal Cell. *Journal of Mathematical Imaging and Vision*, 21(3):235–279, November 2004. ISSN 0924-9907. doi: 10.1023/B:JMIV.0000043739.51886.01.

Anderson, T. W. On the distribution of the two-sample Cramer-von Mises criterion. *The Annals of Mathematical Statistics*, 33(3):1148–1159, 1962.

Anderson, T. W. and Darling, D. A. Asymptotic theory of certain goodness-of-fit criteria based on stochastic processes. *The Annals of Mathematical Statistics*, 23(2):193–212, 1952.

Backman, S., Maekynen, A. J., Kolehmainen, T. T., and Ojala, K. M. Fast lens testing using random targets. *Opto-Ireland 2002: Optics and Photonics Technologies and Applications*, 4876(1):1100–1109, 2003. doi: 10.1117/12.463918.

Backman, S., Makynen, A., Kolehmainen, T., and Ojala, K. Random target method for fast MTF inspection. *Optics Express*, 12:2610–2615, 2004.

Bishop, T., Zanetti, S., and Favaro, P. Light Field Superresolution. In *Proc. ICCP'09. IEEE International Conference on Computational Photography*, 2009. doi: 10.1109/ICCPHOT.2009.5559010.

Bishop, T. E. and Favaro, P. The Light Field Camera: Extended Depth of Field, Aliasing, and Superresolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5):972–986, 2012. ISSN 0162-8828. doi: 10.1109/TPAMI.2011.168.

Bony, J. *Cours d'analyse. Théorie des distributions et analyse de Fourier*. Les Éditions de l'École Polytechnique, 2001.

Bookstein, F. L. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis & Machine Intelligence (PAMI)*, 11(6):567–585, June 1989. doi: 10.1109/34.24792.

Bouguet, J. Y. Camera calibration toolbox for matlab, 2008. `http://www.vision.caltech.edu/bouguetj/calib_doc/`.

Bouman, C. A., Pollak, I., and Wolfe, P. J., editors. *Superresolution with the focused plenoptic camera*, volume 7873, San Francisco, CA, 02/07/2011 2011. SPIE. doi: 10.1117/12.872666.

Bradski, G. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

Brauers, J., Seiler, C., and Aach, T. Direct PSF estimation using a random noise target. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 7537, 2010. doi: 10.1117/12.837591.

Buades, A., Coll, B., Morel, J., et al. A review of image denoising algorithms, with a new one. *SIAM Journal on Multiscale Modeling and Simulation*, 4(2):490–530, 2005.

Capel, D. *Image Mosaicing and Super-Resolution (Cphc/Bcs Distinguished Dissertations)*. SpringerVerlag, 2004. ISBN 1852337710.

Chalmond, B. PSF estimation for image deblurring. *CVGIP: Graphical Models and Image Processing*, 53(4):364 – 372, 1991. ISSN 1049-9652. doi: 10.1016/1049-9652(91)90039-M.

Chaudhuri, S. and Rajagopalan, A. *Depth from defocus: a real aperture imaging approach*. Springer Verlag, 1999.

Cheng, O., Guangzhi, W., Quan, Z., Wei, K., and Hui, D. Evaluating harris method in camera calibration. *Conference Proceedings of the International Conference of IEEE Engineering in Medicine and Biology Society*, 6:6383–6386, 2005.

Choudhury, P. and Tumblin, J. The trilateral filter for high contrast images and meshes. In *Proceedings of the 14th Eurographics workshop on Rendering*, pages 186–196. Eurographics Association, 2003.

Claxton, C. D. and Staunton, R. C. Measurement of the point-spread function of a noisy imaging system. *Journal of the Optical Society of America A (JOSA A)*, 25(1):159–170, Jan 2008. doi: 10.1364/JOSAA.25.000159.

Coulange, B. and Moisan, L. An aliasing detection algorithm based on suspicious colocalizations of Fourier coefficients. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 2013–2016. IEEE, September 2010. ISBN 978-1-4244-7992-4. doi: 10.1109/ICIP.2010.5651195.

Dammertz, H., Sewtz, D., Hanika, J., and Lensch, H. P. A. Edge-avoiding À-trous wavelet transform for fast global illumination filtering. In *Proceedings of the Conference on High Performance Graphics*, HPG '10, pages 67–75, Aire-la-Ville, Switzerland, Switzerland, 2010. Eurographics Association.

Daniels, A., Boreman, G., Ducharme, A., and Sapir, E. Random transparency targets for modulation transfer function measurement in the visible and infrared regions. *Optical Engineering*, 34(3):860 – 868, 1995.

Delbracio, M., Almansa, A., and Musé, P. Recovering the Subpixel PSF from Two Photographs at Different Distances. *Image Processing On Line*, preprint, 2012a.

Delbracio, M., Musé, P., and Almansa, A. Non-parametric Sub-pixel Local Point Spread Function Estimation. *Image Processing On Line*, 2012, 2012b. doi: 10.5201/ipol.2012.admm-nppsf.

Dutré, P., Bala, K., and Bekaert, P. *Advanced global illumination, 2nd edition*. A K Peters (http://www.akpeters.com/), 2006.

Egan, K., Tseng, Y.-T., Holzschuch, N., Durand, F., and Ramamoorthi, R. Frequency analysis and sheared reconstruction for rendering motion blur. *ACM Trans. Graph.*, 28(3):93:1–93:13, July 2009. ISSN 0730-0301. doi: 10.1145/1531326.1531399.

Egan, K., Durand, F., and Ramamoorthi, R. Practical filtering for efficient ray-traced directional occlusion. *ACM Trans. Graph.*, 30(6):180:1–180:10, December 2011a. ISSN 0730-0301. doi: 10.1145/2070781.2024214.

Egan, K., Hecht, F., Durand, F., and Ramamoorthi, R. Frequency analysis and sheared filtering for shadow light fields of complex occluders. *ACM Trans. Graph.*, 30(2):9:1–9:13, April 2011b. ISSN 0730-0301. doi: 10.1145/1944846.1944849.

Gastal, E. S. L. and Oliveira, M. M. Adaptive manifolds for real-time high-dimensional filtering. *ACM Trans. Graph.*, 31(4):33:1–33:13, July 2012. ISSN 0730-0301. doi: 10.1145/2185520.2185529.

Goodman, J. W. *Introduction To Fourier Optics.* McGraw-Hill, 2nd edition, 1996. ISBN 0070242542.

Grant, M. and Boyd, S. CVX: Matlab software for disciplined convex programming (web page and software), September 2012. `http://cvxr.com/cvx`.

Grompone von Gioi, R., Jakubowicz, J., Morel, J.-M., and Randall, G. LSD: a Line Segment Detector. *Image Processing On Line*, 2012, 2012. doi: 10.5201/ipol.2012.gjmr-lsd.

Hachisuka, T., Jarosz, W., Weistroffer, R. P., Dale, K., Humphreys, G., Zwicker, M., and Jensen, H. W. Multidimensional adaptive sampling and reconstruction for ray tracing. *ACM Trans. Graph.*, 27:33:1–33:10, August 2008. ISSN 0730-0301. doi: 10.1145/1360612.1360632.

Harris, C. and Stephens, M. A combined corner and edge detector. In *Proceedings of the Fourth Alvey Vision Conference*, pages 147–151, 1988.

Healey, G. and Kondepudy, R. Radiometric CCD camera calibration and noise estimation. *IEEE Transactions on Pattern Analysis & Machine Intelligence (PAMI)*, 16(3):267–276, March 1994. ISSN 0162-8828. doi: 10.1109/34.276126.

Hörmander, L. *The analysis of linear partial differential operators. I. Distribution theory and Fourier analysis.* Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]. Springer-Verlag, Berlin, 1983. ISBN 3-540-12104-8.

ISO. ISO 12233:2000: Photography – Electronic still-picture cameras – Resolution measurements, 2000.

Jensen, H. W. and Christensen, N. J. Optimizing path tracing using noise reduction filters. In *Proceedings of WSCG*, 1995.

Joshi, N., Szeliski, R., and Kriegman, D. PSF Estimation using Sharp Edge Prediction. In *IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, Alaska, June 2008.

Kajiya, J. T. The rendering equation. *SIGGRAPH Comput. Graph.*, 20:143–150, August 1986. ISSN 0097-8930. doi: 10.1145/15886.15902.

Ladjal, S. *Flou et quantification dans les images numériques.* PhD thesis, Centre de Mathématiques et de Leurs Applications, Ecole Normale Supérieure de Cachan, March 2005.

LEBRUN, M., COLOM, M., BUADES, A., AND MOREL, J. Secrets of image denoising cuisine. *Acta Numerica*, 21(1):475–576, 2012.

LEE, J.-S. Digital image smoothing and the sigma filter. *Computer Vision, Graphics, and Image Processing*, 24(2):255 – 269, 1983. ISSN 0734-189X. doi: 10.1016/0734-189X(83)90047-6.

LEE, M. E. AND REDNER, R. A. Filtering: A note on the use of nonlinear filtering in computer graphics. *IEEE Comput. Graph. Appl.*, 10:23–29, May 1990. ISSN 0272-1716. doi: 10.1109/38.55149.

LEHTINEN, J., AILA, T., CHEN, J., LAINE, S., AND DURAND, F. Temporal light field reconstruction for rendering distribution effects. *ACM Trans. Graph.*, 30(4):55:1–55:12, July 2011. ISSN 0730-0301. doi: 10.1145/2010324.1964950.

LEHTINEN, J., AILA, T., LAINE, S., AND DURAND, F. Reconstructing the indirect light field for global illumination. *ACM Trans. Graph.*, 31(4):51:1–51:10, July 2012. ISSN 0730-0301. doi: 10.1145/2185520.2185547.

LEVIN, A., FERGUS, R., DURAND, F., AND FREEMAN, W. T. Image and depth from a conventional camera with a coded aperture. *ACM Trans. Graph.*, 26(3), July 2007. ISSN 0730-0301. doi: 10.1145/1276377.1276464.

LEVY, E., PELES, D., OPHER-LIPSON, M., AND LIPSON, S. Modulation transfer function of a lens measured with a random target method. *Applied Optics*, 38(4):679 – 683, 1999.

LLC, I. Imatest 3.6. `http://www.imatest.com/`, 2010.

LOWE, D. G. Object recognition from local scale-invariant features. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1150–1157 vol.2, Los Alamitos, CA, USA, August 1999. IEEE Computer Society. ISBN 0-7695-0164-8. doi: 10.1109/ICCV.1999.790410.

LUCCHESE, L. AND MITRA, S. K. Using saddle points for subpixel feature detection in camera calibration targets. In *APCCAS (2)*, pages 191–195. IEEE, 2002. doi: 10.1109/APCCAS.2002.1115151.

LUXEN, M. AND FÖRSTNER, W. Characterizing image quality: Blind estimation of the point spread function from a single image. In *Proceedings of Photogrammetric Computer Vision 2002*, pages 205–210, 2002.

MARION, A. *Acquisition et Visualisation des Images*. Eyrolles, 1997. ISBN 2-212-08871-X.

MCCOOL, M. D. Anisotropic diffusion for Monte Carlo noise reduction. *ACM Trans. Graph.*, 18: 171–194, April 1999. ISSN 0730-0301. doi: 10.1145/318009.318015.

MOISAN, L., MOULON, P., AND MONASSE, P. Automatic Homographic Registration of a Pair of Images, with A Contrario Elimination of Outliers. *Image Processing On Line*, 2012, 2012. doi: 10.5201/ipol.2012.mmm-oh.

MOREL, J. AND LADJAL, S. Notes sur l'analyse de Fourier et théorie de Shannon en traitement d'images. In *Analyse de Fourier et traitement dimages. Journées X-UPS*, pages 37–100, Ecole Polytechnique, F91128 Palaiseau cedex, France., 1998. http://www.math.polytechnique.fr/xups/.

NG, M., CHAN, R., AND TANG, W. A fast algorithm for deblurring models with neumann boundary conditions. *SIAM Journal on Scientific Computing*, 21(3):851–866, 2000.

Overbeck, R. S., Donner, C., and Ramamoorthi, R. Adaptive wavelet rendering. *ACM Trans. Graph.*, 28:140:1–140:12, December 2009. ISSN 0730-0301. doi: 10.1145/1618452.1618486.

Paris, S., Kornprobst, P., Tumblin, J., and Durand, F. A gentle introduction to bilateral filtering and its applications. In *ACM SIGGRAPH 2007 courses*, SIGGRAPH '07, New York, NY, USA, 2007. ACM. doi: 10.1145/1281500.1281602.

Park, S., Park, M., and Kang, M. Super-resolution image reconstruction: a technical overview. *Signal Processing Magazine, IEEE*, 20(3):21–36, 2003.

Pharr, M. and Humphreys, G. *Physically Based Rendering, Second Edition: From Theory To Implementation*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2nd edition, 2010. ISBN 0123750792, 9780123750792.

Portugal, L. F., Júdice, J. J., and Vicente, L. N. A comparison of block pivoting and interior-point algorithms for linear least squares problems with nonnegative variables. *Mathematics of Computation*, 63(208):625–643, 1994. ISSN 0025-5718. doi: 10.2307/2153286.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 3 edition, 2007. ISBN 0521880688, 9780521880688.

Raskar, R., Agrawal, A., and Tumblin, J. Coded exposure photography: motion deblurring using fluttered shutter. *ACM Trans. Graph.*, 25(3):795–804, July 2006. ISSN 0730-0301. doi: 10.1145/1141911.1141957.

Reichenbach, S. E., Park, S. K., and Narayanswamy, R. Characterizing digital image acquisition devices. *Optical Engineering*, 30(2):170–177, 1991. doi: 10.1117/12.55783.

Robinson, D. and Milanfar, P. Statistical performance analysis of super-resolution. *Image Processing, IEEE Transactions on*, 15(6):1413–1428, June 2006. ISSN 1057-7149. doi: 10.1109/TIP.2006.871079.

Rooms, F., Philips, W., and Portilla, J. Parametric PSF estimation via sparseness maximization in the wavelet domain. In Truchetet, F. and Laligand, O., editors, *Proceedings of the SPIE - Wavelet Applications in Industrial Processing II*, volume 5607, pages 26–33, 2004.

Rousselle, F., Knaus, C., and Zwicker, M. Adaptive sampling and reconstruction using greedy error minimization. *ACM Trans. Graph.*, 30:159:1–159:12, December 2011. ISSN 0730-0301. doi: 10.1145/2070781.2024193.

Rubner, Y., Tomasi, C., and Guibas, L. J. A metric for distributions with applications to image databases. In *Proceedings of the Sixth International Conference on Computer Vision*, ICCV '98, pages 59–66, Washington, DC, USA, 1998. IEEE Computer Society. ISBN 81-7319-221-9.

Rushmeier, H. E. and Ward, G. J. Energy preserving non-linear filters. In *Proceedings of the 21st annual conference on computer graphics and interactive techniques*, SIGGRAPH'94, pages 131–138, New York, NY, USA, 1994. ACM. ISBN 0-89791-667-0. doi: 10.1145/192161.192189.

Sen, P. and Darabi, S. On filtering the noise from the random parameters in Monte Carlo rendering. *ACM Trans. Graph.*, 31(3):18:1–18:15, June 2012. ISSN 0730-0301. doi: 10.1145/2167076.2167083.

Shirley, P., Aila, T., Cohen, J., Enderton, E., Laine, S., Luebke, D., and McGuire, M. A local image reconstruction algorithm for stochastic rendering. In *Proceedings of ACM SIGGRAPH 2011 Symposium on Interactive 3D Graphics and Games*, pages 9–13. ACM Press, 2011.

Smith, E. H. B. PSF estimation by gradient descent fit to the ESF. In *Proceedings of SPIE - Image Quality and System Performance III*, volume 6059, 60590E. SPIE, January 2006. doi: 10.1117/12.643071.

Soler, C., Subr, K., Durand, F., Holzschuch, N., and Sillion, F. Fourier depth of field. *ACM Trans. Graph.*, 28(2):18:1–18:12, May 2009. ISSN 0730-0301. doi: 10.1145/1516522.1516529.

Sprengel, R., Rohr, K., and Stiehl, H. Thin-plate spline approximation for image registration. In *Engineering in Medicine and Biology Society, 1996. Bridging Disciplines for Biomedicine. Proceedings of the 18th Annual International Conference of the IEEE*, volume 3, pages 1190–1191 vol.3, Oct-3 Nov 1996. doi: 10.1109/IEMBS.1996.652767.

Stein, E. and Weiss, G. *Introduction to Fourier Analysis on Euclidean Spaces. (PMS-32)*. Princeton University Press, November 1971. ISBN 069108078X.

Stephens, M. A. Use of the Kolmogorov-Smirnov, Cramér-von Mises and related statistics without extensive tables. *Journal of the Royal Statistical Society Series B*, 32(1):115–122, 1970.

Tian, H., Fowler, B., and Gamal, A. E. Analysis of temporal noise in CMOS photodiode active pixel sensor. *IEEE Journal of Solid-state Circuits*, 36(1):92–101, 2001. doi: 10.1109/4.896233.

Traonmilin, Y., Ladjal, S., and Almansa, A. On the amount of regularization for super-resolution reconstruction. December 2012. `http://hal.archives-ouvertes.fr/hal-00763984`.

Veach, E. *Robust Monte Carlo methods for light transport simulation*. PhD thesis, Stanford University, Stanford, CA, USA, 1997.

Šroubek, F., Cristóbal, G., and Flusser, J. A unified approach to superresolution and multichannel blind deconvolution. *IEEE Transactions on Image Processing*, 16(9):2322 – 2332, 2007. ISSN 10577149.

Wertheimer, M. Untersuchungen zur lehre von der gestalt. ii. *Psychological Research*, 4(1):301–350, 1923.

Wikipedia. Bayer filter — wikipedia, the free encyclopedia, 2012. `http://en.wikipedia.org/w/index.php?title=Bayer_filter&oldid=526941312`. [Online; accessed 12-December-2012].

Williams, C. S. and Becklund, O. A. *Introduction to the Optical Transfer Function (SPIE Press Monograph Vol. PM112)*. SPIE Publications, 2002. ISBN 0819443360.

Xu, Q., Liu, Y., Zhang, R., Bao, S., and Scopigno, R. Noise reduction for path traced imaging of participating media. In *European Signal Processing Conference (Eusipco)*, Barcelona, Spain, September 2011.

Xu, R. and Pattanaik, S. N. A Novel Monte Carlo Noise Reduction Operator. *IEEE Comput. Graph. Appl.*, 25:31–35, March 2005. ISSN 0272-1716. doi: 10.1109/MCG.2005.31.

Yadid-Pecht, O. Geometrical modulation transfer function for different pixel active area shapes. *Optical Engineering*, 39(4):859–865, 2000. doi: 10.1117/1.602462.

Zandhuis, J., Pycock, D., Quigley, S., and Webb, P. Sub-pixel non-parametric PSF estimation for image enhancement. *IEE Proceedings - Vision, Image, and Signal Processing*, 144(5):285–292, 1997. doi: 10.1049/ip-vis:19971307.

Zhang, W. and Cham, W.-K. A single image based blind super-resolution approach. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 329–332, Oct. 2008. doi: 10.1109/ICIP.2008.4711758.

Zhang, Z. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis & Machine Intelligence (PAMI)*, 22(11):1330–1334, 2000. ISSN 0162-8828. doi: 10.1109/34.888718.

Zhao, T., Wang, R., Liu, Y., and Yu, F. Characteristic-analysis of optical low pass filter used in digital camera. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 6034, pages 60340N.1 – 60340N.9. SPIE, 2006. doi: 10.1117/12.668107.