Image interpolation and denoising for division of focal plane sensors using Gaussian Processes

Elad Gilboa,[1] **John P. Cunningham,**[2] **Arye Nehorai,**[1] **and Viktor Gruev** [3]

[1]*Preston M. Green Department of Electrical & Systems Engineering, Washington University in St. Louis, St. Louis, MO 63130*
[2]*Department of Statistics, Columbia University, New York, NY 10032*
[3]*Department of Computer Science Engineering, Washington University in St. Louis, St. Louis, MO 63130*

gilboae@ese.wustl.edu

Abstract: Image interpolation and denoising are important techniques in image processing. These methods are inherent to digital image acquisition as most digital cameras are composed of a 2D grid of heterogeneous imaging sensors. Current polarization imaging employ four different pixelated polarization filters, commonly referred to as division of focal plane polarization sensors. The sensors capture only partial information of the true scene, leading to a loss of spatial resolution as well as inaccuracy of the captured polarization information. Interpolation is a standard technique to recover the missing information and increase the accuracy of the captured polarization information.

Here we focus specifically on Gaussian process regression as a way to perform a statistical image interpolation, where estimates of sensor noise are used to improve the accuracy of the estimated pixel information. We further exploit the inherent grid structure of this data to create a fast exact algorithm that operates in $\mathcal{O}\left(N^{3/2}\right)$(vs. the naive $\mathcal{O}\left(N^3\right)$), thus making the Gaussian process method computationally tractable for image data. This modeling advance and the enabling computational advance combine to produce significant improvements over previously published interpolation methods for polarimeters, which is most pronounced in cases of low signal-to-noise ratio (SNR). We provide the comprehensive mathematical model as well as experimental results of the GP interpolation performance for division of focal plane polarimeter.

OCIS codes: (230.5440) Polarization-selective devices; (330.6180) Spectral discrimination; (280.0280) Remote sensing and sensors; (110.5405) Polarimetric imaging.

References and links

1. N. J., *Image Sensors and Signal Processing for Digital Still Cameras* (Cambridge U press, 2005).
2. E. R. Fossum, "Cmos image sensors: electronic camera-on-a-chip," IEEE Transactions on Electron Devices **44**, 1689–1698 (1997).
3. V. Gruev, Z. Yang, J. Van der Spiegel, and R. Etienne-Cummings, "Current mode image sensor with two transistors per pixel," IEEE Transactions on Circuits and Systems I-Regular Papers **57**, 1154–1165 (2010).
4. R. Njuguna and V. Gruev, "Current-mode cmos imaging sensor with velocity saturation mode of operation and feedback mechanism," IEEE Sensors Journal **14**, 1710–721 (2014).

5. V. Gruev, R. Perkins, and T. Yor, "Ccd polarization imaging sensor with aluminum nanowire optical filters," Optics Express **18**, 19087–19094 (2010).

6. Y. Liua, T. York, W. Akersa, G. Sudlowa, V. Gruev, and S. Achilefua, "Complementary fluorescence-polarization microscopy using division-of-focal-plane polarization imaging sensor," Journal of Biomedical Optics **17**, 116001.1–116001.4 (2012).

7. D. Miyazaki, R. Tan, K. Hara, and K. Ikeuchi, "Polarization-based inverse rendering from a single view," in "Ninth IEEE International Conference on Computer Vision," , vol. 2 (2003), vol. 2, pp. 982–987.

8. Y. Y. Schechner and N. Karpel, "Recovery of underwater visibility and structure by polarization analysis," IEEE J. Oceanic Eng **30**, 570587 (2005).

9. T. Krishna, C. Creusere, and D. Voelz, "Passive polarimetric imagery-based material classification robust to illumination source position and viewpoint," IEEE Trans. Image Process **20**, 288292 (2011).

10. M. Anastasiadou, A. D. Martino, D. Clement, F. Liége, B. Laude-Boulesteix, N. Quang, J. Dreyfuss, B. Huynh, A. Nazac, L. Schwartz, and H. Cohen, "Polarimetric imaging for the diagnosis of cervical cancer," Phys. Status Solidi **5**, 14231426 (2008).

11. A. Giachetti and N. Asuni, "Real-time artifact-free image upscaling," IEEE Transactions on Image Processing **20**, 2760–2768 (2011).

12. D. Zhou, "An edge-directed bicubic interpolation algorithm," 2010 3rd International Congress on Image and Signal Processing (CISP), **3**, 1186–1189 (2010).

13. B. Bayer, "Color imaging array," (1976).

14. B. Ratliff, C. LaCasse, and S. Tyo, "Interpolation strategies for reducing ifov artifacts in microgrid polarimeter imagery," Opt. Express **17**, 9112–9125 (2009).

15. S. Gao and V. Gruev, "Bilinear and bicubic interpolation methods for division of focal plane polarimeters," Opt. Express **19**, 26161–26173 (2011).

16. S. Gao and V. Gruev, "Gradient-based interpolation method for division-of-focal-plane polarimeters," Opt. Express **21**, 1137–1151 (2013).

17. J. Tyo, C. LaCasse, and B. Ratliff, "Total elimination of sampling errors in polarization imagery obtained with integrated microgrid polarimeters," Opt. Lett **34**, 3187–3189 (2009).

18. X. Li and M. T. Orchard, "New edge-directed interpolation," IEEE Transactions on Image Processing **10**, 1521–1527 (2001).

19. R. Hardie, D. LeMaster, and B. Ratliff, "Super-resolution for imagery from integrated microgrid polarimeters," Opt. Express **19**, 12937–12960 (2011).

20. H. He and W. Siu, "Single image super-resolution using Gaussian process regression," in "CVPR, 2011 IEEE Conference on," (IEEE, 2011), pp. 449–456.

21. P. J. Liu, "Using Gaussian process regression to denoise images and remove artifacts from microarray data," Master's thesis, University of Toronto (2007).

22. D. Higdon, "Space and space–time modeling using process convolutions," Quantitative methods for current environmental issues p. 37 (2002).

23. G. Xia and A. E. Gelfand, "Stationary process approximation for the analysis of large spatial datasets," Tech. rep., Duke U (2006).

24. C. K. Winkle and N. Cressie, "A dimension-reduced approach to space-time Kalman filtering," Biometrika **86** (1999).

25. N. Cressie and G. Johannesson, "Fixed rank Kriging for very large spatial data sets," Journal of the Royal Statistical Society: Series B (Statistical Methodology) **70** (2008).

26. A. J. Storkey, "Truncated covariance matrices and Toeplitz methods in Gaussian processes," in "ICANN," (1999).

27. R. Furrer, M. G. Genton, and D. Nychka, "Covariance tapering for interpolation of large spatial datasets," Journal of Computational and Graphical Statistics **15**, 502–523 (2006).

28. J. Quiñonero-Candela and C. Rasmussen, "A unifying view of sparse approximate Gaussian process regression," JMLR **6** (2005).

29. C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning* (The MIT Press, 2006).

30. M. Lázaro-Gredilla, "Sparse Gaussian processes for large-scale machine learning," Ph.D. thesis, Universidad Carlos III de Madrid, Madrid, Spain (2010).

31. H. Rue and H. Tjelmeland, "Fitting Gaussian Markov random fields to Gaussian fields," Scandinavian Journal of Statistics **29** (2002).

32. K. Chalupka, C. K. I. Williams, and I. Murray, "A framework for evaluating approximation methods for Gaussian process regression," JMLR (2013).

33. J. P. Cunningham, K. V. Shenoy, and M. Sahani, "Fast Gaussian process methods for point process intensity estimation," in "ICML," (2008), pp. 192–199.

34. D. Duvenaud, H. Nickisch, and C. E. Rasmussen, "Additive Gaussian processes," in "NIPS," (2011), pp. 226–234.

35. E. Gilboa, Y. Saatçi, and J. P. Cunningham, "Scaling multidimensional Gaussian processes using projected additive approximations," JMLR: W&CP **28** (2013).

36. E. Gilboa, Y. Saatçi, and J. P. Cunningham, "Scaling multidimensional inference for structured Gaussian pro-

cesses," IEEE-TPAMI (2013).

37. R. Perkins and V. Gruev, "Signal-to-noise analysis of Stokes parameters in division of focal plane polarimeters," Optics Express **18** (2010).

38. J. Luttinen and A. Ilin, "Efficient Gaussian process inference for short-scale spatio-temporal modeling," in "JMLR: W&CP," , vol. 22 (2012), vol. 22, pp. 741–750.

39. Y. Luo and R. Duraiswami, "Fast near-GRID Gaussian process regression," JMLR: W&CP **31** (2013).

40. K. E. Atkinson, *An introduction to numerical analysis* (John Wiley & Sons, 2008).

41. M. Fiedler, "Bounds for the determinant of the sum of hermitian matrices," in "Proceedings of the American Mathematical Society," , vol. 30 (1971), vol. 30, pp. 27 – 31.

42. C. Williams and M. Seeger, "Using the Nyström method to speed up kernel machines," in "NIPS 13," (2001).

43. P. W. Goldberg, C. K. Williams, and C. M. Bishop, "Regression with input-dependent noise: A Gaussian process treatment," NIPS **10** (1998).

44. A. G. Wilson, D. A. Knowles, and Z. Ghahramani, "Gaussian process regression networks," ICML (2012).

1. Introduction

Solid state imaging sensors, namely CMOS and CCD cameras, capture two of the three fundamental properties of light: intensity and color. The third property of light, polarization, has been ignored by CMOS and CCD sensor used for daily photography [1, 2, 3, 4]. However, In nature, many species are capable of sensing polarization properties of light in addition to intensity and color. The visual system in these species combines photoreceptors and specialized optics capable of filtering the polarization properties of the light field. Recent development in nanofabrication and nano-photonics has enabled the realization of compact and high resolution polarization sensors. These sensors, known as division of focal plane polarimeters (DoFP), monolithically integrate pixelated metallic nanowire filters, acting as polarization filters, with an array of imaging elements [5]. One of the main advantages of division-of-focal-plane sensors compared to division-of-time sensors is the capability of capturing polarization information at every frame. The polarization information captured by this class of sensors can be used to extract various parameters from an imaged scene, such as microscopy for tumor margin detection [6], 3-D shape reconstruction from a single image [7], underwater imaging [8], material classification [9], and cancer diagnosis [10].

The idea of monolithically integrating optical elements with an array of photo sensitive elements is similar to today's color sensors, where a Bayer color filter is integrated with an array of CMOS/CCD pixels. The monolithic integration of optics and imaging elements have allowed for realization of compact color imaging sensors. Since pixelated color filters are placed on the imaging plane, spatial resolution is lost up to 4 times for the red and blue channel and 2 times for the green channel. Interpolation algorithms for color image sensors have been constantly improving over the last three and half decades, allowing for improvements in color replication accuracy and recovery of the lost spatial resolution [11, 12]. An important reason for the wide acceptance of color imaging technologies is the utilization of interpolation algorithms. Since pixelated color filters are placed on the imaging plane, each sensor only observes partial information (one color), and thus it is standard practice to interpolate missing components across the sensors. Cameras using division of focal plane polarization sensors utilize four pixelated polarization filters, whose transmission axis is offset by 45 degree from each other (see Fig. 1). Hence, each pixel represents only 1/4 of the linear polarization information, severely lowering the resolution of the image and the accuracy of the reconstructed polarization information.[1] Interpolation techniques allow to estimate the missing pixel orientations across the imaging array, thereby improving both the accuracy of polarization information and mitigate the loss of spatial resolution.

---

[1] Although the four pixels are spatially next to each other, their instantaneous field of view is different, resulting in different intensities. The pixels intensities are used to reconstruct the polarization information, such as Stokes vector, angle and degree of linear polarization.
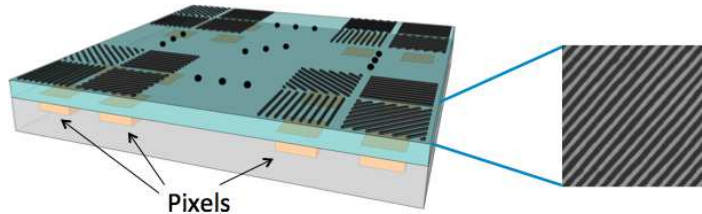
Fig. 1. Division of focal plane polarization sensors on the imaging plane. Each pixel captures only 1/4 of the polarization information. Interpolation is needed for recovering the full camera resolution.

Similar problems were encountered in color imaging sensors, when the Bayer pattern for pixelated color filters was introduced in the 1970s [13]. In order to recover the loss of spatial resolution in color sensors and improve the accuracy of the captured color information, various image interpolation algorithms have been developed in the last 30 years. For DoFP, the common use is of conventional image interpolation algorithms such as bilinear, bicubic, and bicubic-spline, which are based on space-invariant non-adaptive linear filters [14, 15, 16]. Frequency domain filtering was introduced for band limited images and total elimination of the reconstruction error was achieved [17]. More advanced algorithms use adaptive algorithms, such as the new edge-directed interpolation (NEDI) [18], utilize the multi-frame information, such as [19]. Recently, there has been a growing interest in the use of GP regression for interpolation and denoising of image data for color sensors [20, 21]. GP is a Bayesian nonparametric statistical method that is a reasonable model for natural image sources [18], and fits well the DoFP interpolation setting by accounting for heteroscedastic noise and missing data, as we will discuss next.

Whereas conceptually attractive, exact GP regression suffers from $\mathcal{O}(N^3)$ runtime for data size N, making it intractable for image data (N is the number of pixels which is often on the order of millions for standard cameras). Many authors have studied reductions in complexity via approximation methods, such as simpler models like kernel convolution [22, 23], moving averages [24], or fixed numbers of basis functions [25]. A significant amount of research has also gone into sparse approximations, including covariance tapering [26, 27], conditional independence to inducing inputs [28, 29], sparse spectrum [30], or a Gaussian Markov random field approximation [31]. While promising, these methods can have significant approximation penalties [28, 32]. A smaller number of works investigate reduction in complexity by exploiting special structure in the data, which avoids the accuracy/efficiency tradeoff at the cost of generality. Examples of such problems are: equidistant univariate input data where the fast Fourier transform can be used (e.g., [33]); additive models with efficient message passing routines (e.g., [34, 35, 36]); and multiplicative kernels with multidimensional grid input [36], as we discuss here.

Image data is composed of multiple pixels that lie on a two dimensional grid. The multidimensional grid input data induces exploitable algorithmic structures for GP models with multiplicative kernels [36]. [2] We show in Sec. 2 that these conditions, along with the assumption of homogenous noise, admit an efficient algorithm, we named GP-grid, that can be used to significantly lower the computational cost while still achieving high accuracy.

---

[2]GP models with multiplicative kernels and multidimensional grid inputs are constrained to have (i) covariance structure that separates multiplicatively across its D inputs (see Section 2); and (ii) a restricted input space: input points $\mathbf{x} \in \mathbf{X}$ can only lie on a multidimensional grid $\mathbf{X} = \mathscr{X}^1 \times \mathscr{X}^2 \times \ldots \times \mathscr{X}^D \subset \mathbb{R}^D$, where $\mathscr{X}^d = \{x_1^d, x_2^d, \ldots, x_{|\mathscr{X}_d|}^d\}$ is the input space of dimension d, $1 \leq d \leq D$.

However, there are two important cases where two key assumptions break down. First, the input might not lie on a complete grid. This can often occur due to missing values (e.g., malfunctioning sensors), or when analyzing irregular shaped segment of the image. Second, captured image data often contain heteroscedastic signal-dependent noise [37]. In fact, the heteroscedastic nature of the data is often overlooked by most of the image interpolation techniques in the literature and can significantly reduce the interpolation accuracy (see Sec. 3). Thus, the second contribution of this work is to utilize the actual noise statistics of the data acquisition system, which fits naturally into the GP framework. With these advances, it is possible to significantly improve both the interpolation and denoising performance over current methods.

In Sec. 2.1 we extend the GP-grid algorithm to handle two limitations of the basic algorithm by allowing for (i) incomplete data, and (ii) heteroscedastic noise. Certainly these extensions to standard GP have been used to good purpose in previous GP settings, but their success can not be replicated in the large N case without additional advances related to this specific multidimensional grid structure.

The main contribution of this paper is to present an efficient GP inference for improved interpolation for DoFP polarimeters. The GP statistical inference is able learn the properties of the data and incorporates an estimation of the sensor noise in order to increase the accuracy of the polarization information and improve spatial resolution.

1.1. Gaussian Process Regression

Interpolation and denoising are essentially problems of regression where we are interested in learning the underline true image from the noisy and partial observed data of the captured image. Gaussian processes offer a powerful statistical framework for regression that is very flexible to the data. Here we will briefly introduce the GP framework and the main equations which carry the computational burden of the method.

In brief, GP regression is a Bayesian method for nonparametric regression, where a prior distribution over continuous functions is specified via a Gaussian process (the use of GP in machine learning is well described in [29]). A GP is a distribution on functions $\mathbf{f}$ over an input space $\mathbf{X}$ (For 2 dimensional image data case $\mathbf{X} = \mathbb{R}^2$) such that any finite selection of input locations $\mathbf{x}_1, \dots, \mathbf{x}_N \in \mathbf{X}$ gives rise to a multivariate Gaussian density over the associated targets, i.e.,

$$p(f(\mathbf{x}_1), \dots, f(\mathbf{x}_N)) = \mathcal{N}(\mathbf{m}_N, \mathbf{K}_N), \tag{1}$$

where $\mathbf{m}_N = m(\mathbf{x}_1, \dots, \mathbf{x}_N)$ is the mean vector and $\mathbf{K}_N = \{k(\mathbf{x}_i, \mathbf{x}_j; \theta)\}_{i,j}$ is the covariance matrix, for mean function $m(\cdot)$ and covariance function $k(\cdot, \cdot; \theta)$. Throughout this work, we use the subscript N to denote that $\mathbf{K}_N$ has size $N \times N$. We are specifically interested in the basic equations for GP regression, which involve two steps. First, for given data $\mathbf{y} \in \mathbb{R}^N$ (making the standard assumption of zero-mean data, without loss of generality), we calculate the predictive mean and covariance at L unseen inputs as

$$\mu_L = \mathbf{K}_{LN} \left(\mathbf{K}_N + \sigma_n^2 \mathbf{I}_N \right)^{-1} \mathbf{y}, \tag{2}$$

$$\Sigma_L = \mathbf{K}_L - \mathbf{K}_{LN} \left(\mathbf{K}_N + \sigma_n^2 \mathbf{I}_N \right)^{-1} \mathbf{K}_{NL}, \tag{3}$$

where σ_n^2 is the variance of the observation noise (the *homogeneous* noise case, where σ_n^2 is constant across all observations), which is assumed to be Gaussian. Because the function $k(\cdot, \cdot; \theta)$ is parameterized by hyperparameters θ such as amplitude and lengthscale, we must also consider the log marginal likelihood $Z(\theta)$ for model selection:

$$\log Z(\theta) = -\frac{1}{2} \left(\mathbf{y}^\top (\mathbf{K}_N + \sigma_n^2 \mathbf{I}_N)^{-1} \mathbf{y} + \log |\mathbf{K}_N + \sigma_n^2 \mathbf{I}_N| + N \log(2\pi) \right). \tag{4}$$

Here we use this marginal likelihood to optimize over the hyperparameters in the usual way [29]. The runtime of exact GP regression (henceforth "full-GP") regression and hyperparameter learning is $\mathcal{O}(N^3)$ due to the $\left(\mathbf{K}_N + \sigma_n^2 \mathbf{I}_N\right)^{-1}$ and $\log|\mathbf{K}_N + \sigma_n^2 \mathbf{I}_N|$ terms. The significant computational and memory costs make GP impractical for many applications such as image analysis where the number of data points can easily reach the millions. Next, we will show how we can exploit the inherent structure of our problem setting to allow for very fast computations, making GP an attractive method for image interpolation and denoising.

2. Fast GP for Image Data

In this Section we present algorithms which exploit the existing structure of the image data for significant savings in computation and memory, but with the same exact inference achieved with standard GP inference (using Cholesky decomposition).

Gaussian process inference and learning requires evaluating $(\mathbf{K} + \sigma^2 \mathbf{I})^{-1}\mathbf{y}$ and $\log|\mathbf{K} + \sigma^2 \mathbf{I}|$, for an $N \times N$ covariance matrix $\mathbf{K}$, a vector of N datapoints $\mathbf{y}$, and noise variance σ^2, as in Equations (2) and (4), respectively. For this purpose, it is standard practice to take the Cholesky decomposition of $(\mathbf{K} + \sigma^2 \mathbf{I})$ which requires $\mathcal{O}(N^3)$ computations and $\mathcal{O}(N^2)$ storage, for a dataset of size N. However, in the context of image data, there is significant structure on $\mathbf{K}$ that is ignored by taking the Cholesky decomposition.

In [36], it was shown that exact GP inference with multiplicative kernel[3] can be done in $\mathcal{O}(N^{\frac{3}{2}})$ operations (compared to the standard $\mathcal{O}(N^3)$ operations) for input data that lie on a grid. The computation stemmed from the fact that:

1. $\mathbf{K}$ is a Kronecker product of 2 matrices (a *Kronecker matrix*) which can undergo eigen-decomposition with only $\mathcal{O}(N)$ storage and $\mathcal{O}(N^{\frac{3}{2}})$ computations

2. The product of Kronecker matrices or their inverses, with a vector $\mathbf{u}$, can be performed in $\mathcal{O}(N^{\frac{3}{2}})$ operations.

Given the eigendecomposition of $\mathbf{K}$ as $\mathbf{Q}\mathbf{V}\mathbf{Q}^\top$, we can re-write $(\mathbf{K} + \sigma^2 \mathbf{I})^{-1}\mathbf{y}$ and $\log|\mathbf{K} + \sigma^2 \mathbf{I}|$ in Eqs. (2) and (4) as

$$(\mathbf{K} + \sigma^2 \mathbf{I})^{-1}\mathbf{y} = (\mathbf{Q}\mathbf{V}\mathbf{Q}^\top + \sigma^2 \mathbf{I})^{-1}\mathbf{y} \tag{5}$$
$$= \mathbf{Q}(\mathbf{V} + \sigma^2 \mathbf{I})^{-1}\mathbf{Q}^\top \mathbf{y}, \tag{6}$$

and

$$\log|\mathbf{K} + \sigma^2 \mathbf{I}| = \log|\mathbf{Q}\mathbf{V}\mathbf{Q}^\top + \sigma^2 \mathbf{I}| = \sum_{i=1}^{N} \log(\lambda_i + \sigma^2), \tag{7}$$

where λ_i are the eigenvalues of $\mathbf{K}$, which can be computed in $\mathcal{O}(N^{\frac{3}{2}})$.

Thus we can evaluate the predictive distribution and marginal likelihood in Eqs. (2) and (4) to perform *exact* inference and hyperparameter learning, with $\mathcal{O}(N)$ storage and $\mathcal{O}(N^{\frac{3}{2}})$ operations.

Unfortunately, for images of focal plane sensors, the above assumptions - (1) a grid-complete dataset with (2) homogeneous noise $\sigma_n^2 \mathbf{I}_N$ - are too simplistic, and will cause GP-grid to substantially underfit, resulting in degraded estimation power of the method (this claim will be substantiated later in Sec. 3). First, incomplete grids often occur due to missing values (e.g., malfunctioning sensors), or non-grid input space (e.g., a segment of an image). Second, the

---

[3]multiplicative kernel is defined as $k(x_i, x_j) = \prod_{d=1}^{2} k^d(x_i^d, x_j^d)$.

homogeneous noise assumption is not valid a sensor noise is input-dependent and often can be represented by a linear model which is camera dependent [37].

In the next section we will extend GP-grid to efficiently handle the case of $\mathbf{K} + \mathbf{D}$, where $\mathbf{K}$ is not a Kronecker matrix, and $\mathbf{D}$ is a positive diagonal matrix of known heteroscedastic noise.

2.1. Inference

Incomplete grids often occur due to missing values (e.g., malfunctioning sensors), or non-grid input space (e.g., a segment of an image). Previous work in literature tried to handel missing observation for grid input using either sampling [38], or a series of rank-1 updates [39]; however, both of these methods incur high runtime cost with the increase of the number of missing observations.

We will use the notation $\mathbf{K}^M$ to represent a covariance matrix that was computed using multiplicative kernel over an input set χ^M, $|\chi^M| = M$, which do need to lie on a complete grid. Hence, $\mathbf{K}^M$ is not necessarily a non Kronecker matrix, and can be represented as $\mathbf{K}^M = \mathbf{E}\mathbf{K}^N\mathbf{E}^\top$, where the $\mathbf{K}^N$ is a Kronecker matrix computed from the set χ^N ($\chi^M \subseteq \chi^N$) of N inputs that lie on a complete grid. The $\mathbf{E}$ matrix is a selector matrix of size $M \times N$, choosing the inputs from the complete-grid space that are also in the incomplete-grid space. The $\mathbf{E}$ matrix is a sparse matrix having only M non-zero elements.

This representation is helpful for matrix vector multiplication because it allows us to project the incomplete observation vector to the complete grid space $\mathbf{y}^N = \mathbf{E}^\top\mathbf{y}^M$, perform all operations using the properties of the Kronecker matrices, and then project the results back to the incomplete space.

We use preconditioned conjugate gradients (PCG) [40] to compute $(\mathbf{K}^M + \mathbf{D})^{-1}\mathbf{y}$. Each iteration of PCG calculates the matrix vector multiplication

$$(\mathbf{K}^M + \mathbf{D})\mathbf{v} = \mathbf{E}\mathbf{K}^N\mathbf{E}^\top\mathbf{v} + \mathbf{D}\mathbf{v}. \tag{8}$$

The complexity of matrix vector multiplication of the diagonal and selector matrices is $\mathcal{O}(N)$, hence the complexity of the multiplication above will depend on the matrix vector multiplication of $\mathbf{K}^N$. Exploiting the fast multiplication of Kronecker matrices, PCG takes $\mathcal{O}(JN^{\frac{3}{2}})$ total operations (where the number of PCG iterations $J \ll N$) to compute $(\mathbf{K}^M + \mathbf{D})^{-1}\mathbf{y}$, which allows for exact inference.

2.2. Learning

For learning (hyperparameter training) we must evaluate the marginal likelihood of Eq. (4). We cannot efficiently compute the complexity penalty in the marginal likelihood $\log |\mathbf{K}^M + \mathbf{D}|$ because $\mathbf{K} = \mathbf{K}^M + \mathbf{D}$ is not a Kronecker matrix. We can alleviate this problem by replacing the exact logdet complexity with an efficient upperbound. Using an upperbound allows to keep the computational and memory complexities low while maintaining a low model complexity. We emphasize that only the log determinant (complexity penalty) term in the marginal likelihood undergoes a small approximation, and inference remains exact.

In [41], the author showed that for a $n \times n$ hermitian positive semidefinite matrices $\mathbf{A}$, $\mathbf{B}$ with eiganvalues $\alpha_1 \leq \alpha_2 \leq \ldots \leq \alpha_n$ and $\beta_1 \leq \beta_2 \leq \ldots \leq \beta_n$, respectively,

$$|\mathbf{A} + \mathbf{B}| \leq \prod_{i=1}^{n} (\alpha_i + \beta_{n+1-i}). \tag{9}$$

These estimates are best possible in terms of the eigenvalues of $\mathbf{A}$ and $\mathbf{B}$. Using Eq. (9), we can

write an upperbound on the complexity penalty as:

$$\log |\mathbf{K}^M + \mathbf{D}| \leq \sum_{i=1}^{M} \log \left(\lambda_i^M + d_{M+1-i} \right), \qquad (10)$$

where $d_i = \text{sort}(\text{diag}(\mathbf{D}))_i$. However, finding λ_i^M, the eigenvalues of $\mathbf{K}^M$, is still $\mathscr{O}\left(N^3\right)$. We instead approximate the eigenvalues λ_i^M using the eigenvalues of $\mathbf{K}^N$, such that $\tilde{\lambda}_i^M = \frac{M}{N}\lambda_i^N$ for $i = 1, \ldots, M$ [42], which is a particularly good approximation for large M (e.g., $M > 1000$).

3. Results

Here we show that GP-grid allows for improved accuracy results for division of focal plane polarimeter compared to other commonly-used interpolation methods.

3.1. Runtime Complexity

First, we compare the runtime complexity of GP-grid from Section 2.1 to both full-GP (naive implementation of Sec. 1.1 using Cholesky decomposition) and GP-grid with grid-complete and homogeneous noise. We conduct the comparison using a segment of real image data of a cone (Fig. 2). We consider only the input locations within the segment (size M), except for GP-grid homogeneous where we used the entire grid-complete segment (size N). At each iteration the size of the window N is increased, thereby increasing the number of input locations M (pixels we did not mask out). Fig. 2 illustrates the time complexity of the three algorithms as a function of input size (pixels). For every comparison we also note the ratio of unmasked input to the total window size for each point. The time complexity presented for all algorithms is for a single calculation of the negative log marginal likelihood (NLML) and its derivatives (dNLML), which are the needed calculations in GP learning (and which carry the complexity of the entire GP algorithm). In GP-grid, the noise model is not learned but assumed to be known from the apparatus used to capture the image [37], which is:

$$\sigma_i^2 = 0.47I_i + 56.22, \qquad (11)$$

where at location i, σ_i^2 is the noise variance and I_i is the image intensity. For simplicity, we assume a general model for all the sensors in the imaging plane. Since we do not have I_i we use the measured y_i instead as an approximation, which is a common heuristic (though it technically violates the generative model of the GP) of known camera properties that we discuss later in this work. As can be seen in Fig. 2, GP-grid does inference exactly and scales only superlinearly with the input size, while full-GP is cubic. While the more general GP-grid (Sec. 2.1) does slightly increase computational effort, it does so scalably while preserving exact inference, and we will show that it has significant performance implications that easily warrant this increase. All other commonly used interpolation methods (e.g., bilinear, bicubic, and bicubic-spline) scale at least linearly with the data.

3.2. Application to Division of Focal Plane Images

In this section we test GP-grid on real division of focal plane image data, and we demonstrate improvement in accuracy of polarization (Stokes) parameters compared to commonly-used methods. For better comparison we used four different scenes, each captured multiple times using (*i*) a short shutter speed resulting in low signal-to-noise ratio (SNR) images, and (ii) a long shutter speed resulting in high SNR images. We acquired hundreds of images for each scene using a CCD imaging array (Kodak KAI-4022 4MP) and a polarization filter. We used four polarization filters corresponding to angles: 0, 45, 90, and 135 (see Fig. 3).
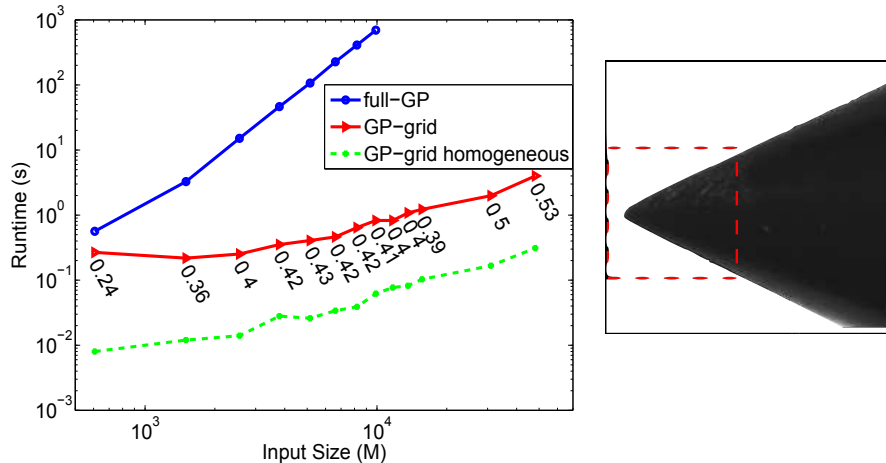
Fig. 2. Runtime complexity of full-GP, GP-grid, and GP-grid homogeneous, for a single calculation of the negative log marginal likelihood (NLML) and its derivatives. For input, we used segmented data from the cone image of the right. At every comparison the size of the segment N (red dotted line) was increased, thereby increasing the input size M (pixels not masked out). The ratio of input size to the complete grid size (M/N) is shown next to the GP-grid plot. The slope for the full-GP is 2.6, for GP-grid is 1.0, and for GP-grid homogeneous is 1.1 (based on the last 8 points). This empirically verifies the improvement in scaling. Other interpolation methods also scale at least linearly, so the cost of running GP-grid is constant (the runtime gap is not widening with data).

To extract the noiseless ground-truth (the basis of our comparisons), we averaged over the majority of the pictures, holding out a small subset for testing. In order to test the interpolation performance, we interpolated the entire image using only a subset of the image (downsampled by four). All images are around 40000 pixels, hence, even their down-sampled version will be impractical for the standard naive GP implementation. The interpolated images were then compared to their corresponding averaged images for accuracy analysis. The accuracy criterion we used was the normalized mean square error (NMSE) between the interpolated images and the average images, defined as:

$$\text{NMSE}(y, \bar{y}) = \frac{\frac{1}{N} \sum_i^N (y_i - \bar{y}_i)^2}{var(\bar{y})}, \tag{12}$$

where $\bar{y}$ is the data of averaged image.[4] Normalization is used in order to compare between the results of the low and high SNR images since they have a different intensity range. We compare GP-grid with the common interpolation algorithms: bilinear, bicubic, bicubic-spline (Bic-sp), NEDI [18], and frequency domain (FD) filter [17]. [5] Although this is by no means an exhaustive comparison, it does allow for a benchmark for comparison with GP performance. Note that in all the comparisons we intentionally discarded the border pixels (five pixels width) so they would be most favorable to non-GP methods as the non-GP methods fail particularly badly at the edges. Had we included the border pixels, our GP-grid algorithm would perform even better in comparison to conventional methods.

---

[4]If we consider $\bar{y}$ to be our signal, the NMSE can be seen as an empirical inverse of the SNR.

[5]The FD filter parameters were chosen such that 95% of the Stokes parameters spectrum of the averaged images was captured.
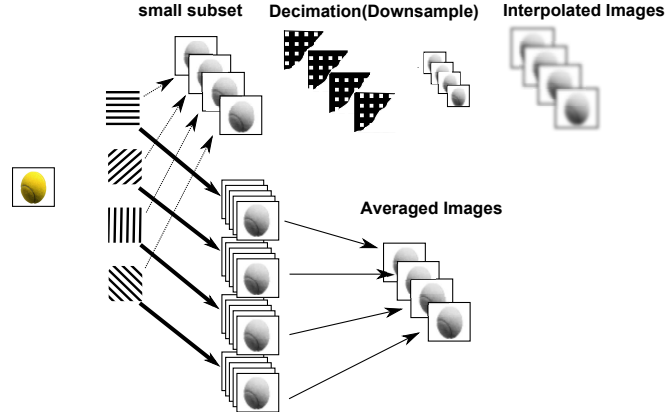
Fig. 3. The original image on the left is passed through four polarization filters with different phases. Over a hundred filtered images are captured. A small subset of the filtered images is used for the interpolation testing and the rest are averaged to approximate the noiseless filtered images. The filtered images used for testing are downsampled by four (using different downsampling patterns) and then interpolated back to original size.

We explore real data using our improved GP-grid model. Performance of course depends critically on the noise properties of the system, which in captured images is primarily sensor noise. Other works in the literature consider additional GPs to infer a heteroscedastic noise model [43, 44], which brings additional computational complexity that is not warranted here. Instead, the simple model of Eq. (11) works robustly and simply for this purpose. We ran GP-grid using a multiplicative Matérn($\frac{1}{2}$) covariance function, and learned the hyperparameters: lengthscales (l_1, l_2), signal variance (σ_f^2) [29].

In this section, reconstruction errors are presented from a set of three different images. The images are segmented into a background image and foreground image, where the foreground image is the object of interest such as the horse (Fig. 6) or toy (Fig. 7. Reconstruction is performed on both the foreground and background images separately, as well as the entire image. Segmenting the object of interest from the image and applying the six different interpolation methods avoids reconstruction errors on the edge boundary between the object and background. Different illumination conditions were considered for the same scene, effectively emulating different SNR conditions. In total, six different images are analyzed and the normalized squared mean error is reported. The images chosen for this analysis have both high and low frequency component in the S_0, S_1 and S_2 image and allow for analysis of images that would be similar to real-life images.

The first set of results presented is for the "Mug" scene (Fig. 4). The scene is composed of a bright mug in front of a bright background. The brightness of the images is important as a brighter image will produce higher luminance and a higher signal in the camera. The top row of the figure shows a summary of the results for short shutter speed images and the bottom row shows the results for long shutter speed images. As can be expected, the intensity range of the low SNR test image on the top is much lower than the high SNR test image on the bottom. Also, we can see that the normalized error distribution in the low SNR image is significantly higher than the high SNR image. Following the scheme presented in Fig. 3, we used the interpolated and averaged images to compute the Stokes parameters

$$S_0 = I_0 + I_{90}, \qquad S_1 = I_0 - I_{90}, \qquad S_2 = I_{45} - I_{135}. \tag{13}$$

In the right side of Fig. 4 we show a comparison of the normalized error between the Stokes

parameters calculated using the interpolated images (using different interpolation methods) and the Stokes parameters calculated from the averaged images. The bar plots allow for easy comparisons between the six interpolation methods for each of the Stokes parameters. Note that the low SNR and high SNR cases must be considered separately since they use different scaling. It is clear that GP outperformed all the other methods in this scene for each of the Stokes parameters. The results of the computed Stokes parameters for the mug scene of Fig. 4 are illustrated in Fig. 5. Fig. 5 shows the dominance of noise in the reconstruction of common interpolation algorithms for low SNR images. We averaged 10000 images to produce a ground truth image where the effective noise is decreased by a factor of around 100 from a single captured image. The GP interpolation achieves significantly better polarization accuracy in the low SNR case compared to the other five interpolation methods. The improvement is most evident for the S_1 and S_2 parameters since they are differentiation operators which are more susceptible to noise. The frequency domain filtering methods proposed in [17], has the worst reconstruction performance due to the fact that the images used for this example are not band limited. Hence, the S_1 and S_2 images cannot be easily filtered out using a non-adaptive Butterworth filter.

A similar comparison was done for the three additional scenes: Horse (Fig. 6), and Toy (Fig. 7). Differently than the Mug scene analysis, here we manually separated the comparison for the object and the background. The reason for the separation is because the two segments have very different properties (spatial frequencies) and learning on the entire image will result in a kernel that will be suboptimal on each region separately. Close analysis of the results show two important facts. First, the improvement was higher for low SNR images than high SNR images. This is not surprising as all the interpolation methods are excepted to perform well when the noise level is low compared to the signal level.

Second, S_1 and S_2 show higher improvement compared to S_0 image. This is because the S_0 image by construction is less sensitive to noise due to the averaging of several pixels in a given neighborhood, while S_1 and S_2 are significantly more sensitive to noise due to taking the difference between neighboring pixels. In other words, the S_0 image has higher SNR compared to S_1 and S_2 images and in-depth mathematical modeling of the SNR can be found in [37]. Improving the accuracy of S_1 and S_2 are especially important because of their nonlinear dependent to the other polarization parameters: angle of polarization $\phi = (1/2)\tan^{-1}(S_2/S_1)$ and the degree of linear polarization $\rho = \sqrt{S_1^2 + S_2^2}/S_0$.
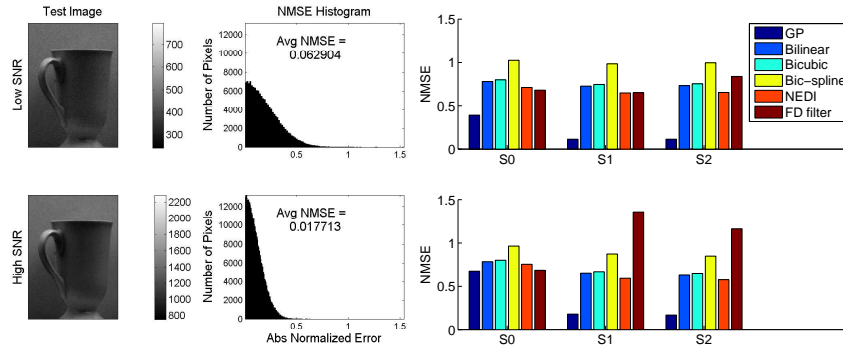

Fig. 4. Left column shows the noisy test image before decimation (subsampling) and interpolation. Middle column shows the histogram of the absolute normalized error and the average NMSE for the captured noisy image compared to the average image. The Stokes parameters comparison is shown on the right for the six interpolation methods tested. Comparison between the six interpolation methods should be considered for each of the Stokes parameters separately.
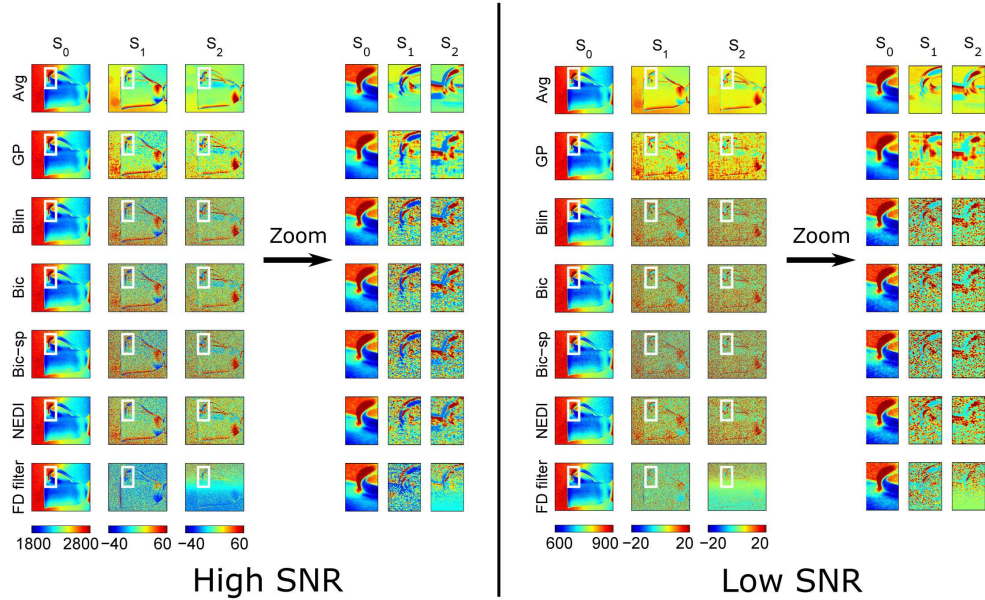
Fig. 5. Results of the Stokes parameters for the different interpolation methods of the mug scene for low and high SNR. The total NMSE of the methods is summarized in Fig. 4. The left (right) panel shows the results for a high (low) SNR image. The white box indicates the zoom-in region for each of Stokes parameters results. The first row shows the Stokes parameters computed on temporally averaged images, which we use as the underline ground truth. The following rows show the results for the rest of the interpolation methods.

4. Conclusion

GP allows for statistical interpolation that can naturally incorporate the camera noise model. Overall, the results of our experiments show that the GP framework allows for improved reconstructions of the Stokes parameters over conventional interpolation methods. Improvement was most evident in low SNR images where the recovering of S_1 and S_2 is most difficult, and where having a good prior can help reduce the effect of the noise.

Another interesting realization that came out of the comparison presented in this paper is that the Bicubic-spline algorithm performance greatly degrades in the presence of noise. This result is different than other papers in the literature where the comparison was done on the averaged images only [15, 16]. The spectral method of [17] was also suboptimal, which is likely because our tested scenes where not band limited, and the effect of input dependent noise on the spectrum.

GP becomes tractable for image data by using the GP-grid algorithm we introduce here, and it is a convenient technology to naturally incorporate our two performance-critical advances: segmentation (incomplete grids) and a known noise model. As the results show, all of these advances are important in order for GP to be considered a general framework for image data.

It is common practice in image processing to mix different methods in order to improve the overall results, e.g., alternate methods close to an edge. Integrating GP-grid together with other state-of-the-art interpolation methods to achieve further improvement is an interesting topic for future work.
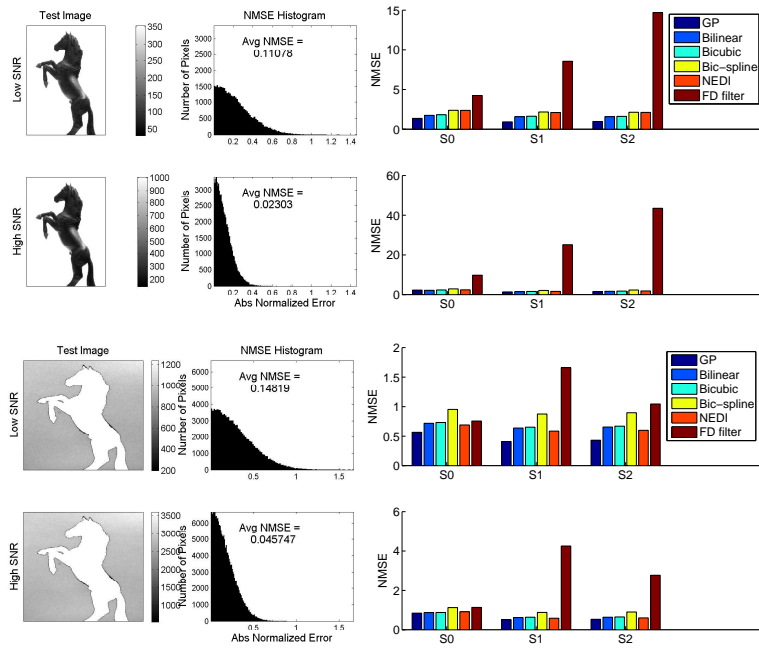
Fig. 6. This figure illustrates the results for the segmented horse scene i.e. the toy horse is segmented from the background and interpolation is only performed on the toy horse portion of the image. The first two rows show results for the horse object when discarding the information of the background. The white pixels indicate locations that where not used in the analysis. The bottom two rows show results when performing interpolation on the background part of the image, i.e. excluding the horse from the scene. The left column shows the noisy test images, middle column shows the histogram of the absolute normalized error, and the right column shows a comparison between the six different interpolation methods tested.
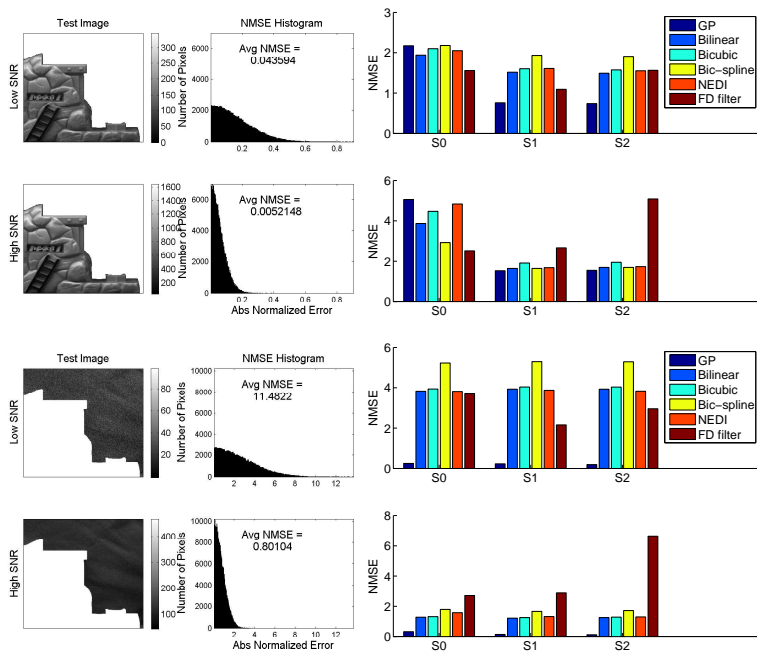
Fig. 7. Toy Scene. See caption of Fig. 6.