“A COMPUTATIONAL APPROACH TO FOURIER SYNTHESIS FROM NON-UNIFORM SAMPLES”

VICENTE LIZANA ESTIVILL

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN INFORMÁTICA

Profesor Guía: Dr. Mauricio Araya López
Profesor Correferente: Dr. Claudio Torres López

Febrero - 2020
A mi padre Rolando, quien me dió a mi y a mi familia las oportunidades que hemos tenido, con mucho esfuerzo y sacrificio. A mi madre Patricia y mi hermana Guadalupe, quienes me estimularon desde una edad temprana promoviendo la gestación de mi pensamiento crítico, y a mi hermano Pablo, de quien continúo aprendiendo en diferentes aspectos.
ABSTRACT

Gaussian Process Regression Image Synthesis is a proposed numerical method intended for image reconstruction via Inverse Fourier Transform directly from a partial non-uniform sampling of the Fourier Space. The motivation for this comes from the field of radio interferometry, where the measurements are obtained within the Fourier space and part of the algorithm for the reconstruction of the images relies on altering the samples in order to meet the restrictions of the Inverse Fast Fourier Transform.

In this work a proposed GPR based algorithm is described, a prototype of it is implemented and then empirically evaluated against the current FFT-based approach. The results of these experiments are positive enough to expect that, with further research and development, this method can become a competitive contender in this field.

KEYWORDS

Radio Interferometry, Image Synthesis, Fourier Transform, Gaussian Process Regression
GLOSSARY

FT: Fourier Transform.
FFT: Fast Fourier Transform.
ALMA: Atacama Large Millimeter Array.
NFFT, NUFFT: Non-Uniform Fast Fourier Transform.
FS: Fourier Series.
IFT: Inverse Fourier Transform.
DFT: Discrete Fourier Transform.
IDFT: Inverse Discrete Fourier Transform.
GPR: Gaussian Process Regression.
AIPY: Astronomical Interferometry in Python.
MSE: Mean Squared Error.
WMSE: Weighted Mean Squared Error.
SNR: Signal to Noise Ratio.
RMS: Root Mean Square.
Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>iii</td>
</tr>
<tr>
<td>KEYWORDS</td>
<td>iii</td>
</tr>
<tr>
<td>GLOSSARY</td>
<td>iv</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>vii</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>viii</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>CHAPTER 1: DEFINITION OF THE PROBLEM</td>
<td>2</td>
</tr>
<tr>
<td>1.1 About the problem</td>
<td>2</td>
</tr>
<tr>
<td>1.1.1 Numerical Analysis and Fourier Transform</td>
<td>2</td>
</tr>
<tr>
<td>1.1.2 Radio interferometry and the Fast Fourier Transform</td>
<td>2</td>
</tr>
<tr>
<td>1.1.3 Non-Uniform Fast Fourier Transform</td>
<td>3</td>
</tr>
<tr>
<td>1.2 Objectives</td>
<td>5</td>
</tr>
<tr>
<td>1.2.1 General Objective</td>
<td>5</td>
</tr>
<tr>
<td>1.2.2 Specific Objectives</td>
<td>5</td>
</tr>
<tr>
<td>CHAPTER 2: CONCEPTUAL FRAMEWORK</td>
<td>6</td>
</tr>
<tr>
<td>2.1 Fourier Transform, Analysis and Synthesis</td>
<td>6</td>
</tr>
<tr>
<td>2.2 Discrete Fourier Transform</td>
<td>7</td>
</tr>
<tr>
<td>2.2.1 Nyquist Sampling Theorem</td>
<td>8</td>
</tr>
<tr>
<td>2.3 Radio Interferometry Imaging</td>
<td>9</td>
</tr>
<tr>
<td>2.4 Gaussian Process Regression</td>
<td>10</td>
</tr>
<tr>
<td>2.4.1 Conditional Gaussian Distribution</td>
<td>12</td>
</tr>
<tr>
<td>2.5 GPR and Image Synthesis</td>
<td>14</td>
</tr>
<tr>
<td>CHAPTER 3: PROPOSED SOLUTION</td>
<td>16</td>
</tr>
<tr>
<td>3.1 Function Generation</td>
<td>16</td>
</tr>
<tr>
<td>3.2 Image synthesis</td>
<td>20</td>
</tr>
<tr>
<td>3.2.1 Synthesis via FFT</td>
<td>20</td>
</tr>
<tr>
<td>3.2.2 Synthesis via GPR</td>
<td>21</td>
</tr>
<tr>
<td>3.3 Metrics</td>
<td>21</td>
</tr>
<tr>
<td>CHAPTER 4: VALIDATION OF THE SOLUTION</td>
<td>23</td>
</tr>
<tr>
<td>4.1 FFT synthesis and metrics</td>
<td>23</td>
</tr>
<tr>
<td>4.2 Exploration of GPR Hyper-parameters</td>
<td>25</td>
</tr>
<tr>
<td>4.3 Optimization of the kernel length</td>
<td>31</td>
</tr>
<tr>
<td>4.4 Noisy samples</td>
<td>31</td>
</tr>
<tr>
<td>4.5 Lambda exploration in noisy samples</td>
<td>36</td>
</tr>
</tbody>
</table>
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Disposition of ALMA telescopes.</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>Simulation of the discrete sampling of Fourier space covered by ALMA in one night, different colors indicate different observing targets.</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>Visualization of the Fourier Transform domains. Time domain in red on the left axis and Frequency domain in blue in the right axis.</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>Diagram of components in radio interferometry imaging.</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>Effect of the $\lambda$ hyper-parameter in GPR. Top: sampled sinc function with random normal noise. Middle: Gaussian Process Regression of the samples, $\lambda \sim 0$. Bottom: Gaussian Process Regression of the samples, $\lambda \sim \sigma^2$ (variance of the noise).</td>
<td>13</td>
</tr>
<tr>
<td>6</td>
<td>Diagram of the experiments basic unit. In blue the images in the physical space, in green the images in Fourier space, in red the measuring criteria and the extracted experiment data in purple.</td>
<td>17</td>
</tr>
<tr>
<td>7</td>
<td>Examples of the functions used in this work. On the left a Gaussian function ($f_G$), on the right an Exponential Decay ($f_E$).</td>
<td>17</td>
</tr>
<tr>
<td>8</td>
<td>Grid of 9 different images rendered with the described configuration.</td>
<td>19</td>
</tr>
<tr>
<td>9</td>
<td>Left: original synthetic image, Right: side by side comparison of the analytical FT vs the FFT (in logarithmic scale).</td>
<td>19</td>
</tr>
<tr>
<td>10</td>
<td>Imaging example. Left: original image, center: FFT reconstruction, right: GPR reconstruction.</td>
<td>20</td>
</tr>
<tr>
<td>11</td>
<td>Visualization of the weights in the WMSE ($k = 1.5$).</td>
<td>22</td>
</tr>
<tr>
<td>12</td>
<td>Boxplot of MSE (top) and WMSE with $k = 2$ (bottom) comparison of visibilities (dashed mean).</td>
<td>24</td>
</tr>
<tr>
<td>13</td>
<td>Heatmap representing the mean of the MSE for the different samples of each combination of hyper-parameters. Top: 15% of visibility, bottom: 30% of visibility. White section shows where GPR is outperformed by FFT.</td>
<td>27</td>
</tr>
<tr>
<td>14</td>
<td>Heatmap representing the mean of the WMSE for the different samples of each combination of hyper-parameters. Top: 15% of visibility, bottom: 30% of visibility. White section shows where GPR is outperformed by FFT.</td>
<td>28</td>
</tr>
</tbody>
</table>
15 Boxplots for image MSE, different values of \( l \) (15% of visibility on top, 30% of visibility on bottom, dashed mean). FFT boxplot is replicated on both sides to avoid misconceptions. ................................. 29

16 Boxplots for image WMSE, different values of \( l \) (15% of visibility on top, 30% of visibility on bottom, dashed mean). FFT boxplot is replicated on both sides to avoid misconceptions. ................................. 30

17 Mean and standard deviation (shaded) of the MSE (top) and WMSE (bottom) for the different values of \( l \). .................................................. 32

18 Examples of GPR reconstruction with 15% of coverage and non-optimal kernel length, first: \( l = 0.2 \), second: \( l = 1.1 \). .................................................... 33

19 Heatmap representing the mean of the MSE (top) and WMSE (bottom) for the different samples of each combination \( \lambda / \text{SNR} \). ........................................... 35

20 FFT reference results for the different values of SNR. ........................................ 36

21 Heatmap representing the mean of the MSE (top) and WMSE (bottom) for the different samples of each combination \( \lambda / \text{SNR} \). Second iteration. ................. 37

22 Mean and standard deviation of the MSE (top) and WMSE (bottom) for FFT and GPR (\( \lambda = 2^{-9} \)) for different SNR values ........................................ 38

23 Three different examples of visibility simulations ................................................. 40

24 Example of synthesis with astronomical-like visibility set. Top left: original image, top right: coverage of the uv-plane, bottom left: FFT reconstruction, bottom right: GPR reconstruction (\( \lambda = 10^{-3}, l = 0.5 \)) ......................................... 40

25 Boxplot for MSE and WMSE of astronomical-like visibility simulations (dashed mean). ................................................................. 41

26 Diagram of the qualitative experiment on the real image.................................... 41

27 Picture of Coyoya the cat, reconstructed from sampling of 30% of Fourier space using \( l = 0.1 \) (top) and \( l = 0.01 \) (bottom). ........................................... 42

List of Tables

1 Results of MSE comparison of visibilities ......................................................... 25

2 Results of WMSE (\( k = 2 \)) comparison of visibilities ..................................... 25
INTRODUCTION

The motivation of this work comes from the field of astronomical radio interferometry, which consists on using several antennas or radio telescopes to gather electromagnetic waves from a single source in the sky. This data is then correlated between each pair of antennas in order to extract the interference patterns, producing a partial surface in Fourier space. The numerical method on which this work is based upon, focuses on the process of retrieving an image from these samples. This is known as Imaging or Image Synthesis.

This document begins with further explanation of the problem stated above, presenting the current approach to tackle image synthesis, stating the general and specific objectives and thus outlining the scope of the work. Following this overview, a conceptual framework is introduced, giving the reader an insight of the different topics this project is based on, including the different kinds of Fourier transforms, sampling, Gaussian Process Regression, and application-related topics, radio interferometry and image synthesis.

The following chapter is about the proposed solution, including the set up of the experiments and the procedurally generated images they use as input, as well as the implementation of the method and the performance comparisons. Next, is the validation of this proposal, where all the experiments performed are described along with the exposition of their results and partial conclusions. The document ends with the general conclusions of the work, the appendix and the references.
CHAPTER 1
DEFINITION OF THE PROBLEM

1.1 About the problem

1.1.1 Numerical Analysis and Fourier Transform

One of the core lines in Scientific Computing is the development of numerical methods and algorithms which can provide sufficiently good solutions to complex mathematical problems, with a small bounded error or no error at all. This error is often the price to pay for a major improvement in computation time and in some cases it is the only way the problem can be handled. Fourier analysis and synthesis are widely used tools in the study of periodic functions, like signals or waves. These have been numerically implemented in algorithms such as the Fast Fourier Transform (FFT), which is an implementation of the Discrete Fourier Transform (DFT) with better time complexity than the direct approach, as it will be analyzed in other sections of this document.

The Fourier Transform has many applications, from its original purpose in the study of Partial Differential Equations\(^1\), to its use in Fractional Calculus applied to Quantum Mechanics [Namias, 1980]. Between its most common applications is signal processing, which includes the study of all kinds of signals (electronic, electromagnetic, mechanical and others) and spectroscopy, which is the study of the interactions between matter and electromagnetic radiation.

The Fast Fourier Transform is a computational implementation of the Discrete version of the Fourier Transform, and despite being an efficient numerical method, imposes strong constraints, making it unsuitable for some specific applications. In particular, it requires a regular discretization of the domains, which is not always possible or under the observer’s control.

1.1.2 Radio interferometry and the Fast Fourier Transform

While spectroscopy is used in several earthly applications domain, like Nuclear Magnetic Resonance Imaging [Sekihara et al., 1984] or Magnetic Resonance Spectroscopic Imaging, the most iconic application for the purposes of this work comes from the field of astronomy.

Radio astronomy is the branch of astronomy that observes the sky in the radio and submillimeter spectra, this is electromagnetic radiation with much larger wavelengths than vis-
The advantages of observing in these spectra include little impact of the atmosphere in the quality of the image, besides detecting phenomena that is not available on other spectra. One of the disadvantages of radio astronomy is that the longer the wavelength, the lower is the resolution the telescope achieves, as the angular resolution \( R \) of the telescopes behaves the following way [Sella, 2012]:

\[
R \propto \frac{\lambda}{D},
\]

where \( \lambda \) is the wavelength and \( D \) is the diameter of the telescope.

In order to achieve a significant increase in resolution without having to significantly increase the diameter of the telescopes, multiple telescopes can be aimed at the same feature. These signals are then merged by feeding them to an instrument called correlator, which extracts the interference patterns between each pair of antennas, leaving measurements in Fourier space. This setup is known as an Interferometer and its angular resolution behaves as [Sella, 2012]:

\[
R \propto \frac{\lambda}{B},
\]

where \( B \) is the distance between the two telescopes that are further away, which is known as the largest baseline between the telescopes.

This poses a new problem. In some observatories, like the Atacama Large Millimeter Array, located in San Pedro de Atacama, Chile, the antennas that compose the interferometer are not regularly placed over the surface, as shown in Figure 1. Furthermore, in an interferometer there is one measure for every baseline, thus, as the earth moves throughout the night, the different measures from the same baseline are not spatially taken from the same relative place. This yields a fully non-uniform sampling of Fourier space as shown in Figure 2.

The process of generating the image from the correlated data (Imaging) roughly consists on obtaining the Inverse Fourier Transform of the measured function. The classical approach to solve this problem is to build an artificial grid from the samples by using gridding algorithms [Casassus et al., 2015], and then performing deconvolutions on the resulting image after the synthesis.

1.1.3 Non-Uniform Fast Fourier Transform

In general, algorithms based on the Fast Fourier Transform from non-uniform samples rely on some homogenization of the samples, and are referred to as Non-Uniform Fast Fourier Transforms, NFFTs or NUFTs [Greengard and Lee, 2004].

---

\(^2\)The spectrum studied by radio astronomy can go from frequencies of 3 Hz. up to 3.000 GHz, this is, from wavelengths of the order of millimeters to kilometers.

\(^3\)Most common deconvolution used is the CLEAN algorithm [Cornwell, 2009].
Figure 1: Disposition of ALMA telescopes.
Source: High Resolution Interferometry in ALMA [Casassus et al., 2015]

Figure 2: Simulation of the discrete sampling of Fourier space covered by ALMA in one night, different colors indicate different observing targets.
Source: casaguides.nrao.edu
There have been attempts at proposing an alternative pipeline to perform Fourier Synthesis, without pre-processing the data to produce an artificial equispaced vector of samples, but working directly with the original samples instead, which would also remove the need for further processing the result. Some examples of this are based on Maximum Entropy Methods \cite{Gull and Daniell, 1978} \cite{Narayan and Nityananda, 1986} \cite{Sutton and Wandelt, 2006} and Gaussian Process Regression. This work aims to validate a specific ongoing research based on the latter, by measuring its performance against the currently used FFT-based method \cite{Thompson et al., 1986}. Other methods were not tried because they are mostly theoretical or in their early stages, there is no public access to code prototypes if existent and the implementation of these would be too complex for the scope of this work.

1.2 Objectives

1.2.1 General Objective

To propose and validate a computational Fourier synthesis method which computes the resulting function directly from non-uniform samples, without fixing them to an equispaced format and minimizing further processing.

1.2.2 Specific Objectives

- To develop a proof of concept of the method in code, in order to perform experiments.
- To achieve a general approach, that solves the problem in one dimension but can be extended to a higher dimensionality if needed.
- To find a suitable measuring criteria for the error, in order to compare the achieved method with respect to the current pipeline.
- To compare the performance of the method with respect to the current pipeline over synthetic data.
CHAPTER 2
CONCEPTUAL FRAMEWORK

2.1 Fourier Transform, Analysis and Synthesis

On 1807, French mathematician Jean-Baptiste Joseph Fourier submitted his paper *Mémoire sur la Propagation de la Chaleur dans les Corps Solides* [Fourier, 1808], or Memoir on the Propagation of Heat in Solid Bodies, where the idea of modeling any function as a trigonometric series was born and later called Fourier Analysis. Later on 1822, in his book *Théorie analytique de la chaleur* [Fourier, 1822] or Analytical Theory of the Heat, the Fourier Transform (FT) was introduced and in time became one of the most important tools in the analysis and synthesis of most kinds of signals.

The definition of the Fourier Series (FS) of a periodic function $f$ is the following:

$$f(x) = s_\infty(x) = a_0 + \sum_{n=0}^{\infty} (a_n \cos(2\pi nx/T) + b_n \sin(2\pi nx/T)),$$

where $T$ is the period of the function and

$$a_0 = \frac{1}{T} \int_{x_0}^{x_0+T} f(x) \, dx,$$

$$a_n = \frac{2}{T} \int_{x_0}^{x_0+T} f(x) \cdot \cos\left(\frac{2\pi nx}{T}\right) \, dx,$$

$$b_n = \frac{2}{T} \int_{x_0}^{x_0+T} f(x) \cdot \sin\left(\frac{2\pi nx}{T}\right) \, dx.$$

The Fourier Transform is a generalization of the Fourier Series defined as follows:

$$\hat{f}(\hat{x}) = \mathcal{F}_x[f(x)] = \int_{-\infty}^{\infty} f(x) \cdot \exp(-2\pi i x \hat{x}) \, dx,$$

where $f$ is the original function of the independent variable $x$ and $i^2 = -1$. When it comes to signals, the FT takes a signal from its original time domain to its representation in the frequency domain (blue), while the Inverse FT (IFT) does the opposite. For example, Figure 3 shows how a signal as a function of time (red), which is composed by a sum of sine waves, is zero on most of the frequency domain except for the peaks in the frequencies of the sine waves, where the height of the peak corresponds to the amplitude of the particular sine wave in that frequency.
Figure 3: Visualization of the Fourier Transform domains. Time domain in red on the left axis and Frequency domain in blue in the right axis. Source: tex.stackexchange.com [Jake, 2013]

The Inverse Fourier Transform is defined as:

\[ f(x) = \mathcal{F}_x^{-1}[\hat{f}(\hat{x})] = \int_{-\infty}^{\infty} \hat{f}(\hat{x}) \cdot \exp(2\pi i x \hat{x}) \, d\hat{x}, \]  

(8)

and the process of generating artificial functions or signals using the IFT is called Fourier Synthesis.

2.2 Discrete Fourier Transform

The Discrete Fourier Transform (DFT) acts over a vector, which might be the discretization of a function or another kind of sequence of numbers. For each element \( x_k \) of the vector \( \vec{x} \), the Discrete Fourier Transform is defined as follows [Sauer, 2012]:

\[ \hat{x}_k = \sum_{n=0}^{N-1} x_n \cdot \exp \left( - \frac{2\pi i}{N} kn \right) \]

(9)

\[ = \sum_{n=0}^{N-1} x_n \left( \cos(2\pi kn/N) - i \cdot \sin(2\pi kn/N) \right), \]

(10)

where \( N \) is the length of the vector. The Inverse Discrete Fourier Transform (IDFT) happens to be a FS in the form:

\[ x_n = \frac{1}{N} \sum_{k=0}^{N-1} \hat{x}_k \cdot \exp \left( \frac{2\pi i}{N} kn \right). \]

(11)

At this point it is fair to notice that the DFT and the IDFT are linear transforms of the form \( \hat{x} = Fx \) and \( x = F^{-1}\hat{x} \), where \( x = (x_0, x_1, \ldots, x_{N-1}) \) and \( \hat{x} = (\hat{x}_0, \hat{x}_1, \ldots, \hat{x}_{N-1}) \). Here,
the DFT Matrix is the following:

\[
F = \begin{pmatrix}
\omega_0^0 & \omega_0^0 & \omega_0^0 & \omega_0^0 & \cdots & \omega_0^0 \\
\omega_0^0 & \omega_1^0 & \omega_2^0 & \omega_3^0 & \cdots & \omega_{n-1}^0 \\
\omega_0^0 & \omega_1^0 & \omega_2^0 & \omega_3^0 & \cdots & \omega_{2(n-1)}^0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
\omega_0^0 & \omega_{(n-1)}^0 & \omega_{2(n-1)}^0 & \omega_{3(n-1)}^0 & \cdots & \omega_{(n-1)(n-1)}^0
\end{pmatrix} 
\in \mathbb{C}^{n \times n},
\]  

(12)

where \( \omega = e^{-2\pi i/N} \), and its inverse follows the property:

\[
F^{-1} = \frac{1}{N} F^*; \tag{13}
\]

because of this, the unitary version of the DFT Matrix is more widely used, of the form:

\[
U = \frac{1}{\sqrt{N}} F; \tag{14}
\]

in order to have a unitary transform that meets \( U^{-1} = U^* \).

### 2.2.1 Nyquist Sampling Theorem

It is important to bear in mind when working with discrete signals that the capability of the sampling to correctly represent a unique signal depends on the sampling frequency. Nyquist sampling theorem states that a band-limited continuous-time signal, with a highest frequency content (bandwidth) of \( B \) Hertz, can be recovered from its samples provided that the sampling frequency \( F_s \) is greater than \( 2B \) samples per second so that there is no aliasing. [Vaseghi, 2007]. This is taken into account in most DFT software libraries to calculate Fourier domain frequencies as follows:

\[
f_{\text{max}} = \frac{n_{\text{samp}}}{4 \cdot t_{\text{max}}}; \tag{15}
\]

where \( f_{\text{max}} \) and \( t_{\text{max}} \) are the boundaries for Fourier and Physical domain respectively and \( n_{\text{samp}} \) the number of samples.

Non-uniform sampling is ruled by an analogous theorem called Whittaker–Shannon–Kotelnikov (WSK) sampling theorem [Marvasti, 2012], which states that a signal can be perfectly reconstructed if the average sampling rate of its samples satisfies Equation (15). Unfortunately, as this theorem applies over periodical signals and also the idea is to perform inference over a surface rather than achieve a perfect reconstruction, further details on this topic fall out of the scope of this work.

---

This equation assumes a symmetric interval in both domains, which suffices for the purposes of this work.
2.3 Radio Interferometry Imaging

In the field of astronomical radio interferometry, there are a few basic components that are quite relevant to this work, depicted in Figure 4. The main concepts are the following:

- **Intensity**: this function denoted by $I(l, m)$ corresponds to the actual image in the physical plane of the object to be observed, it is called intensity because it is a representation of the amplitude of the radio waves captured from the data source.

- **Visibility**: this function denoted by $V(u, v)$ corresponds to the same image but in Fourier space, also named uv-plane for this application. The output of the aforementioned correlators is a partial reconstruction of this surface which comes from the interference pattern between the different antennas, capturing the same wave at different instants of time. Using some form of FT to obtain the intensity from the partial Visibility data is the objective of Image Synthesis.

The data obtained by the correlation of the antennas’ measurements consists on the multiplication between the Visibility and the **Transfer** function $W(u, v)$, which is a sum of Dirac Deltas representing the places of the measurements. As mentioned before, in order to use FFT, this same correlation data must be preprocessed using a Gridding algorithm to achieve a uniform sampling along both $u$ and $v$. The output of the data Gridding process with a spacing of $(\Delta u, \Delta v)$ is the following [Thompson et al., 1986]:

$$V_{\text{meas}}(u, v) = \frac{w(u, v)}{\Delta u \Delta v} \text{III} \left( \frac{u}{\Delta u}, \frac{v}{\Delta v} \right) \{ C(u, v) * [W(u, v)V(u, v)] \},$$

(16)

where the user selected function $C$ is convolved with the measured visibilities to produce a continuous visibility distribution, $w$ is a weighting function\(^5\) and the **Shah** distribution III is defined as follows:

$$\text{III} \left( \frac{u}{\Delta u}, \frac{v}{\Delta v} \right) = \Delta u \Delta v \sum_{i=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \delta(u - i \Delta u, v - k \Delta v).$$

(17)

The imaging process is then complete by the use of the IFT:

$$I_{\text{meas}}(l, m) = \text{III}(l \Delta u, m \Delta v) * \hat{w}(l, m) * \{ \hat{C}(l, m)[\hat{W}(l, m) * I(l, m)] \},$$

(18)

where the hat represents the IFT of a certain function.

\(^5\)The weighting function is intended to correct aspects of the imaging that are out of the scope of this work, e.g. The shape of the antennas. For the purposes of this work this function is going to be uniform.
2.4 Gaussian Process Regression

The Gaussian Process Regression (GPR) is an non-parametric kernel regression method [Bishop, 2006]. First consider a model defined as a linear combination of non-linear space mappings

\[ y(x) = w^T \phi(x), \]  \hfill (19)

or in its vectorial form

\[ y = \Phi w, \]  \hfill (20)

where \( y \) has elements \( y_n = y(x_n) \) and \( \Phi_{n,k} = \phi_k(x_n) \). If a prior distribution

\[ p(w) = \mathcal{N}(0, \alpha I) \]  \hfill (21)

is considered over the vector of weights \( w \), then the probability distribution of \( y \) is also Gaussian with the parameters

\[ \mathbb{E}[y] = \Phi \mathbb{E}[w] = 0, \]  \hfill (22)

\[ \text{cov}(y) = \mathbb{E}[yy^T] = \Phi \mathbb{E}[ww^T] \Phi^T = \alpha \Phi \Phi^T = K, \]  \hfill (23)

where \( \mathbb{E} \) is the expected value of the random variable [Ross, 2014]. Here is where the kernel function arises. Let

\[ k(x, x') = \alpha \phi(x)^T \phi(x'). \]  \hfill (24)
Each element of the covariance matrix is in the form $K_{n,m} = \alpha \phi(x_n)^T \phi(x_m) = k(x_n, x_m)$, which shows that this model can be reformulated into a **Dual Representation** in terms of the kernel function [Bishop, 2006].

It happens that as long as the kernel function complies with being a symmetric positive semidefinite function, the space mapping is not needed to be known, and furthermore it may even take the data to an infinite dimension space without negative implications. This use of the kernel function without the knowledge of the space mapping behind it is known as the **Kernel Trick**.

Now considering a Gaussian noise $\epsilon_n$ in the samples

$$t_n = y_n + \epsilon_n,$$  
(25)

the conditional distribution of the target values with respect to the vector $y$ is

$$p(t|y) = \mathcal{N}(y, \lambda I),$$  
(26)

where $\lambda$ is an hyper-parameter, corresponding to the variance of the noise. As the probability distribution of $y$ is already known from Equations (22) and (23), the marginal distribution for $t$ can be found.

$$p(t) = \mathcal{N}(0, K + \lambda I).$$  
(27)

In order to perform inference over a new point $x_{new}$, notice that

$$p \left( \begin{bmatrix} t \\ t_{new} \end{bmatrix} \right) = \mathcal{N} \left( \begin{bmatrix} 0 \\ K + \lambda I \end{bmatrix}, \begin{bmatrix} k & k^T \\ k(x_{new}, x_{new}) & k^T (K + \lambda I)^{-1} \end{bmatrix} \right),$$  
(28)

where the vector $k$ has elements of the form $k(x_n, x_{new})$. With this, the conditional distribution $p(t_{new}|t)$ turns to be Gaussian with parameters

$$\mu(x_{new}) = k^T (K + \lambda I)^{-1} t,$$  
(29)

$$\sigma^2(x_{new}) = k(x_{new}, x_{new}) - k^T (K + \lambda I)^{-1} k.$$  
(30)

In order to understand better the behavior of the model and the effect of the $\lambda$ hyper-parameter, Figure 5 shows:

- In the first regression (middle), with $\lambda$ close to 0, that the closer $\lambda$ is to 0, the closer the GPR behaves to an interpolation of the values.

---

\[ \text{This is a symmetric function } K : D^2 \to \mathbb{R} \text{ that meets the condition } \sum_{i=1}^n \sum_{j=1}^n c_i c_j K(x_i, x_j) \geq 0, \forall x \in D, c \in \mathbb{R} \]  
[Sauer, 2012]
• In the second regression (bottom), where $\lambda$ coincides with the variance of the noise, a better reconstruction can be appreciated and the shaded area represents a confidence interval of 95% (two standard deviations from the mean, calculated using Equation (30)).

### 2.4.1 Conditional Gaussian Distribution

In order to understand this last result (Equations (29) and (30)) it is important to learn about conditional Gaussian distributions. Let $x$ be a Gaussian distributed vector that can be partitioned in $x_a$ and $x_b$.

\[
p(x) = N(\mu, \Sigma),
\]

\[
p \left( \begin{pmatrix} x_a \\ x_b \end{pmatrix} \right) = N \left( \begin{pmatrix} \mu_a \\ \mu_b \end{pmatrix}, \begin{pmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ab}^T & \Sigma_{bb} \end{pmatrix} \right).
\]

Consider also

\[
\Sigma^{-1} = \Lambda = \begin{pmatrix} \Lambda_{aa} & \Lambda_{ab} \\ \Lambda_{ab}^T & \Lambda_{bb} \end{pmatrix},
\]

then the quadratic form in the exponent of this Gaussian has the form

\[
-\frac{1}{2}(x - \mu)^T \Sigma^{-1} (x - \mu) = -\frac{1}{2}x^T \Sigma^{-1} x + x^T \Sigma^{-1} \mu + c,
\]

where $c$ holds the terms that are independent of $x$. Here the mean and covariance for $p(x_a|x_b)$ can be obtained by coefficient matching with the expression

\[
-\frac{1}{2}(x - \mu)^T \Sigma^{-1} (x - \mu) =
-\frac{1}{2}(x_a - \mu_a)^T \Lambda_{aa} (x_a - \mu_a) - (x_b - \mu_b)^T \Lambda_{ab} (x_b - \mu_b)
\]

\[
-\frac{1}{2}(x_a - \mu_a)^T \Lambda_{aa} (x_a - \mu_a) - (x_b - \mu_b)^T \Lambda_{ab} (x_b - \mu_b).
\]

From the second order terms of $x_a$ it can be concluded that

\[
-\frac{1}{2}x^T \Sigma^{-1} x = -\frac{1}{2}x_a^T \Lambda_{aa} x_a \Rightarrow \Sigma_a|b = \Lambda_{aa}^{-1}.
\]

From the linear terms

\[
x^T \Sigma^{-1} \mu = x_a^T (\Lambda_{aa}\mu_a - \Lambda_{ab} (x_b - \mu_b))
\]

\[
\Rightarrow \mu_a|b = \Sigma_a|b (\Lambda_{aa} \mu_a - \Lambda_{ab} (x_b - \mu_b)) = \mu_a - \Lambda_{aa}^{-1} \Lambda_{ab} (x_b - \mu_b).
\]
Figure 5: Effect of the $\lambda$ hyper-parameter in GPR. Top: sampled \textit{sync} function with random normal noise. Middle: Gaussian Process Regression of the samples, $\lambda \sim 0$. Bottom: Gaussian Process Regression of the samples, $\lambda \sim \sigma^2$ (variance of the noise).

Source: Own elaboration.
Using the properties of the inverse of a partitioned matrix, the values

\[
\Lambda_{aa} = (\Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ab}^T)^{-1}
\]

(38)

\[
\Lambda_{ab} = -(\Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ab}^T)^{-1}\Sigma_{ab}\Sigma_{bb}^{-1}
\]

(39)

can be obtained, thus yielding the desired parameters

\[
\mu_{a|b} = \mu_a + \Sigma_{ab}\Sigma_{bb}^{-1}(x_b - \mu_b)
\]

(40)

\[
\Sigma_{a|b} = \Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ab}^T.
\]

(41)

For simplicity, a generic prior was defined for this proof in Equation (32). By making the connection with the GPR prior in Equation (28), the mean and variance of the GPR in Equations (29) and (30) are obtained. For further details in this topic refer to [Bishop, 2006].

### 2.5 GPR and Image Synthesis

The following idea is unpublished research by the advisor of this thesis and his students. Let \( \mathbf{z}_0 \) be the vector of measured \( \mathbb{R}^2 \) points of the image in Fourier space, or uv-image. For each of these points there is a corresponding complex visibility in the vector \( \mathbf{v}_0 \). As seen before, when adding an unmeasured point to these vectors, the joint probability distribution is

\[
p \left( \left( \mathbf{v}_0 \left| \mathbf{\hat{v}} \right. \right) \right) = \mathcal{N} \left( \begin{pmatrix} K(\mathbf{z}_0, \mathbf{z}_0) + \lambda I & k(\mathbf{z}_0, \mathbf{\hat{z}}) \\ k(\mathbf{\hat{z}}, \mathbf{\hat{z}})^T & k(\mathbf{\hat{z}}, \mathbf{\hat{z}}) \end{pmatrix} \right),
\]

(42)

where \( K(\mathbf{a}, \mathbf{b})_{n,m} = k(\mathbf{a}_n, \mathbf{b}_m) \) and \( k(\mathbf{c}, \mathbf{d})_n = k(\mathbf{c}_n, \mathbf{d}) \). Note that here, in contrast with the previous section, \( \mathbf{\hat{z}} \) is a vector given that it belongs to a two dimensional domain. Using the results from the previous section for the inference of the mean across the uv-image, the Gaussian distribution for \( p(\mathbf{\hat{v}}|\mathbf{v}_0) \) has a mean

\[
\hat{\mu}(\mathbf{z}) = k(\mathbf{z}_0, \mathbf{z})^T(K(\mathbf{z}_0, \mathbf{z}_0) + \lambda I)^{-1}\mathbf{v}_0.
\]

(43)

Taking into account simplicity and the notion that the kernel function should correctly represent the influence of distance between points, a Gaussian kernel was selected, of the form

\[
k(\mathbf{z}, \mathbf{z}') = \exp \left( -\frac{\|\mathbf{z} - \mathbf{z}'\|^2}{2l^2} \right) = \exp \left( -\frac{(\mathbf{z} - \mathbf{z}')^T(\mathbf{z} - \mathbf{z}')}{2l^2} \right),
\]

(44)

where the kernel length \( l \) is introduced as the second hyper-parameter of the model.

The following is the main idea behind this work; As GPR allows an explicit expression for the regression in Fourier space, this expression can be analytically transformed to the physical
space, leaving an analytical closed-form expression for the synthetic image in terms of the original measurements.

Let \( \Psi = (K(z_0, z_0) + \lambda I)^{-1}v_0 \), then

\[
\mu(x) = \mathcal{F}_z^{-1}[\hat{\mu}(z)](x) = \int_{\mathbb{R}^2} k(z_0, z)^T \Psi \exp(2\pi i x^T z) \, dz
\]

\[
= \int_{\mathbb{R}^2} \exp(2\pi i x^T z) \sum_{z' \in z_0} (k(z', z) \cdot \Psi_{z'}) \, dz
\]

\[
= \sum_{z' \in z_0} \Psi_{z'} \int_{\mathbb{R}^2} \exp(2\pi i x^T z) \exp\left(-\frac{||z - z'||^2}{2l^2}\right) \, dz
\]

\[
= \sum_{z' \in z_0} \Psi_{z'} \cdot 2\pi l^2 \exp(2\pi i x^T z') \exp(-2\pi^2 l^2 x^T x)
\]

\[
= 2\pi l^2 \exp(-2\pi^2 l^2 x^T x) \sum_{z' \in z_0} \exp(2\pi i x^T z') \Psi_{z'}
\]

\[
= 2\pi l^2 \exp(-2\pi^2 l^2 x^T x) \zeta^T(K(z_0, z_0) + \lambda I)^{-1}v_0,
\]

where the elements of the vector \( \zeta \) are in the form \( \exp(2\pi i x^T z_0(n)) \). This completes the proposed GPR Synthesis, note that the coefficient \( 2\pi l^2 \) is not relevant for this process given that a normalization is applied to the image afterwards.
CHAPTER 3
PROPOSED SOLUTION

The main idea of this work is to compare the classical approach to astronomical image synthesis with the proposed GPR approach. This is achieved by generating synthetic images, which are later sampled in the Fourier space and synthetized using both methods. The results are compared using an adequate metric.

Each of the experiments detailed in this section are built by iterating over the same basic structure, which is shown in Figure 3. The idea is to explore the potential of the developed GPR Synthesis methods by comparing it to the FFT approach under the same conditions.

3.1 Function Generation

In order to have reasonable astronomical-like images, with a known analytical FT from where to extract the samples, these were composed using two different functions which are typically present in these kinds of images:

- The general bidimensional Gaussian function, defined as follows:
  \[ f_G(x) = A \cdot \exp \left( - (x - \mu)^T \Sigma^{-1} (x - \mu) \right) , \]  
  (52)
  with \( \mu \) being the center of the Gaussian and \( \Sigma^{-1} \) a symmetric positive definite matrix. Its FT is the following:
  \[ \mathcal{F}[f_G](z) = A \pi \sqrt{\det(\Sigma)} \cdot \exp \left( - \pi^2 z^T \Sigma z - 2 \pi i \mu^T z \right) . \]  
  (53)

- The general bidimensional Exponential decay function, defined as follows:
  \[ f_E(x) = A \cdot \exp \left( - \sqrt{(x - \mu)^T \Sigma^{-1} (x - \mu)} \right) = A \cdot \exp \left( - \| M(x - \mu) \| \right) , \]  
  (54)
  with \( \mu \) being the center of the exponential decay and \( \Sigma^{-1} = M^T M \) a symmetric positive definite matrix. Its FT is the following:
  \[ \mathcal{F}[f_E](z) = \frac{2 A \pi \sqrt{\det(\Sigma)}}{(1 + 4 \pi^2 z^T \Sigma z)^{3/2}} \cdot \exp \left( - 2 \pi i \mu^T z \right) . \]  
  (55)

In both cases \( f : \mathbb{R}^2 \to \mathbb{R}, \mathcal{F}[f] : \mathbb{R}^2 \to \mathbb{C} \). The FT of each function was obtained with the aid of the software Wolfram Mathematica [Wolfram Research, 2019] using symbolic integration.

The final image is then a linear combination between two and four of each of these functions, over the domain \( \Omega = [-1, 1]^2 \). The parameters for each function are randomly generated the following way:
Figure 6: Diagram of the experiments basic unit. In blue the images in the physical space, in green the images in Fourier space, in red the measuring criteria and the extracted experiment data in purple.

Source: Own elaboration.

Figure 7: Examples of the functions used in this work. On the left a Gaussian function ($f_G$), on the right an Exponential Decay ($f_E$).

Source: Own elaboration.
• \( A \): Simple bounded random number.

\[
A = r \cdot (B_{\text{upper}}^{(A)} - B_{\text{lower}}^{(A)}) + B_{\text{lower}}^{(A)},
\]

(56)

with \( r \in (0, 1) \) a random number. The values assigned to these bounds were \( B_{\text{upper}}^{(A)} = 0.5 \), \( B_{\text{lower}}^{(A)} = 0.2 \).

• \( \mu \): In order to achieve a better FFT reconstruction, the value of the pixels in the edge of the image must be as close to zero as possible, minimizing the information loss from the function to the image. Because of this the center of the functions is confined to a central region smaller than the whole domain.

\[
\mu = B \cdot (2r_1 - 1, 2r_2 - 1)^T.
\]

(57)

The value for the bound in this case was \( B = 1/3 \).

• \( \Sigma^{-1} \): For this matrix to be symmetric positive definite, a random matrix \( M \) was generated as the result of \( M^T M \) complies with the conditions for any matrix \( M \). This matrix defines the shape of the functions, so it is important to impose boundaries in order to produce significant images. This is done by the following mechanisms:

- Let \( \lambda_1, \lambda_2 \) be the eigenvalues of \( M \) in order of dominance, the ratio between the eigenvalues of \( \Sigma^{-1} \) is bounded by

\[
\left( \frac{\lambda_1}{\lambda_2} \right)^2 < B_{\text{ratio}},
\]

(58)

if the constraint is not met, \( M \) is discarded and re-generated. \( B_{\text{ratio}} = 10 \) was used for the experiments.

- The desired value for the largest eigenvalue \( \rho(\Sigma^{-1}) \) is generated independently from the matrix \( M \) because it defines the widespread of the function, thus by bounding this value the significant part of the surface can be contained within the domain. With this, the matrix is computed as

\[
\Sigma^{-1} = \frac{r \cdot (B_{\text{upper}}^{(\Sigma)} - B_{\text{lower}}^{(\Sigma)}) + B_{\text{lower}}^{(\Sigma)}}{\rho(M)^2} \cdot M^T M,
\]

(59)

with \( r \in (0, 1) \) a random number. In this case \( B_{\text{upper}}^{(\Sigma)} = 60 \), \( B_{\text{lower}}^{(\Sigma)} = 40 \) were used.

As the domain \( \Omega \) and the size of the image are fixed, the sampling rate turns to be

\[
sr = \frac{100}{1 - - 1} = 50,
\]

(60)
Figure 8: Grid of 9 different images rendered with the described configuration. Source: Own elaboration.

Figure 9: Left: original synthetic image, Right: side by side comparison of the analytical FT vs the FFT (in logarithmic scale). Source: Own elaboration.
in both directions, which, according to Nyquist sampling theorem (Equation (15)), imposes a Fourier space domain

$$\Omega_F = [-25, 25]^2.$$  

(61)

In Figure 9 there is an instance of a synthetic image, where it is evident that the FT is computed over the same domain than the one yielded by numpy’s FFT. In this figure it can also be appreciated that while the analytical FT and the FFT of the image share the main characteristics there are slight differences and artifacts, due to FFT’s discrete nature limitations and also its assumption on the image being just a finite window sample of an infinitely periodic function, while in reality the original function is practically zero outside the picture domain.

### 3.2 Image synthesis

As shown in the outline in Figure 6, in order to get any kind of measure and then compare the imaging methods, it is necessary to have a working implementation of both studied methods. There is an example of the outcome of this process in Figure 10.

#### 3.2.1 Synthesis via FFT

For the current FFT approach, the package Astronomical Interferometry in Python (AIPY) [HERA-Team, 2008] was used to perform the synthesis. The image synthesis process consists on rendering the analytical FT of a certain function and selecting a subset of the points in this image, which are passed to the imaging class of this package in order to get a reconstruction in the original domain.
3.2.2 Synthesis via GPR

A prototype of the GPR approach was developed in Python and its scientific computing libraries Numpy and Scipy, the code of the function responsible for the synthesis, gpr_synthesize_image, is available on the Appendix 6.1.

The outline of this function can be summarized as follows:

- It takes as parameters the discrete Fourier surface of the image, the visibility sampling, the desired final imaging size and the hyper-parameters $l$ and $\lambda$.
- It starts by gathering the samples and the corresponding points in Fourier domain, using Numpy’s `fftfreq`.
- It computes the matrix $K + \lambda I$ with the aid of Scipy’s `distance_matrix`.
- The computation of the matrix-vector product $(K + \lambda I)^{-1}v_0$ is done by solving the linear system $(K + \lambda I)x = v_0$.
- For the final part of the computation, three dimensional Numpy arrays are used to store partial results of the expression. Each pixel of the image has one partial result per sample in this part, so the axes of these arrays are the original spacial axes of the image plus an additional axis for the visibilities.
- After obtaining these partial results, the final result is achieved by collapsing these previously mentioned arrays over the visibilities axis, along with adding the already computed components of the expression.

3.3 Metrics

In order to compare the two of these, the first metric to be used is a classic direct approach at measuring the outcome of a regression, which is the **Mean Square Error (MSE)**:

$$
\epsilon_{\text{mse}} = \sum_{i=1}^{N} \sum_{j=1}^{M} (y_{i,j} - \hat{y}_{i,j})^2;
$$

where $y$ represents the original image and $\hat{y}$ represent the synthetized version$^0$. Now for the astronomical application that is the motivation of this work, often not all of the pixels of the image share the same relevance. For this, a **weighted MSE** is proposed:

$$
\epsilon_{\text{wmse}} = \sum_{i=1}^{N} \sum_{j=1}^{M} f(x_{i,j}) \cdot (y_{i,j} - \hat{y}_{i,j})^2.
$$

(63)
Figure 11: Visualization of the weights in the WMSE ($k = 1.5$).
Source: Own elaboration.

From the point of view of the radio interferometry, there are a couple of factors to take into account when choosing this weighting function, for instance, the area of interest is always in the center of the image, and also because of physical conditions like the shape of the antennas, the error tends to increase further away from the center. For these reasons, a Gaussian function was selected for the weighting, which effect on the MSE can be appreciated on Figure 11.

$$\epsilon_{wmse} = \sum_{i=1}^{N} \sum_{j=1}^{M} e^{-kx^T x} \cdot (y_{i,j} - \hat{y}_{i,j})^2.$$  

(64)

\[\text{\textsuperscript{\textdagger}}\text{note that for the purposes of this work, the images will have always the same resolution and } y_{i,j} = y(x_{i,j}).\]
CHAPTER 4
VALIDATION OF THE SOLUTION

The validation of the proposed method consisted on several experiments, covering different aspects of interest of this work. For each experiment, a different number of images were generated, and the variation of this number was mainly due to the computational time required by the experiments. For instance, in some of them only one GPR synthesis per generated image was performed while in others there were several synthesis per generated image, thus even when these latter experiments would run for much longer, in general they would use fewer images.

To ensure that the number of total samples was enough to conclude each experiment, partial results were extracted until there was no longer variation on these results, which would indicate according to the law of large numbers [Dekking et al., 2005] that the experimental expected value was reached.

All the experiments were performed on a laptop running Linux Fedora 30, with 16 gigabytes of RAM and an Intel Core i5-5200U CPU (two cores, four threads, base frequency of 2.2 GHz). The code was written in Python 3.6.7 using the libraries NumPy 1.15.4 and SciPy 1.1.0.

4.1 FFT synthesis and metrics

The first experiment acts as an introduction to the framework, the intention is to validate the correct application of the FFT method as well as the evaluation via the defined metrics. 346 images were used for this experiment, and for each image, 30 random uniform visibility masks were generated, 10 masks covering 15% of the space, 10 masks covering 30% and the last 10 covering 50%. For each visibility, the image was synthetized using FFT method and then evaluated using both metrics, with the results shown in Tables 1, 2 and Figure 12.

Some of the samples were randomly selected and synthetized also using the developed GPR method, with arbitrary hyper-parameters, in order to make the qualitative assertion that both methods were correctly implemented and also validate that the MSE/WMSE values obtained in this experiment could be used as reference values for the next experiments, at least in order of magnitude. From this experiment was concluded that:

- The order of magnitude of the errors, along with the visualization of some of the reconstructions, validate the implementation of both the FFT and GPR synthesis methods.

WMSE with the factor $k = 2$.

The comparison was made in a similar setting as the one shown in Figure 10.
Figure 12: Boxplot of MSE (top) and WMSE with $k = 2$ (bottom) comparison of visibilities (dashed mean).

Source: Own elaboration.
Table 1: Results of MSE comparison of visibilities.

<table>
<thead>
<tr>
<th>Coverage</th>
<th>15%</th>
<th>30%</th>
<th>50%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min</td>
<td>95.23</td>
<td>34.94</td>
<td>13.13</td>
</tr>
<tr>
<td>Q1</td>
<td>381.35</td>
<td>182.91</td>
<td>83.59</td>
</tr>
<tr>
<td>Median</td>
<td>526.04</td>
<td>264.39</td>
<td>129.48</td>
</tr>
<tr>
<td>Mean</td>
<td>572.86</td>
<td>300.13</td>
<td>155.94</td>
</tr>
<tr>
<td>Q3</td>
<td>723.58</td>
<td>378.92</td>
<td>202.48</td>
</tr>
<tr>
<td>Max</td>
<td>2250.68</td>
<td>1118.79</td>
<td>830.8</td>
</tr>
</tbody>
</table>

Source: Own elaboration.

Table 2: Results of WMSE ($k = 2$) comparison of visibilities.

<table>
<thead>
<tr>
<th>Coverage</th>
<th>15%</th>
<th>30%</th>
<th>50%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min</td>
<td>9.88</td>
<td>6.25</td>
<td>2.76</td>
</tr>
<tr>
<td>Q1</td>
<td>74.13</td>
<td>37.73</td>
<td>18.75</td>
</tr>
<tr>
<td>Median</td>
<td>122.09</td>
<td>60.23</td>
<td>32.95</td>
</tr>
<tr>
<td>Mean</td>
<td>163.20</td>
<td>82.06</td>
<td>47.25</td>
</tr>
<tr>
<td>Q3</td>
<td>216.49</td>
<td>99.44</td>
<td>58.7</td>
</tr>
<tr>
<td>Max</td>
<td>981.51</td>
<td>691.05</td>
<td>467.91</td>
</tr>
</tbody>
</table>

Source: Own elaboration.

- The error with respect to the visibility factor seems to be monotonous as expected at least in this percentage range, therefore it is not expected to have significant relative variations in the results by performing the same experiment for several different coverages of Fourier space.

### 4.2 Exploration of GPR Hyper-parameters

The second experiment consisted on an exploration of the GPR hyper-parameters described in Equations (26) and (44).

The first parameter, $\lambda$, is intrinsic to the GPR method and related to the variance of the measurement error of the samples. It is suspected that the impact of this parameter will not be significant for the experiment, as there is no noise being added to the measurements before the synthesis. For this parameter the search space used was

$$B_\lambda = \{10^n\}, \quad n = -9, \ldots, 0.$$  \hfill (65)

The second parameter, $\ell$ or kernel length, belongs to the chosen Gaussian kernel and corresponds roughly to the radius of influence of each one of the measurements. This parameter is suspected to have a much larger impact on the error, as it directly affects the matrices and vectors involved on the GPR inference. Because of this, the search space for this parameter
will have a larger resolution.

\[ B_t = \{0.1 + 0.4n/19\}, \quad n = 0, \ldots, 19. \]  \hfill (66)

For this experiment, a total of 46 images were used. For each image, 10 visibility masks were sampled, 5 with a coverage of 15% and the other 5 with a coverage of 30%. For each of these, the image was first synthesized using FFT as a reference and then using each combination of GPR hyper-parameters \((B_\lambda \times B_t)\).

The results of these experiments are shown in Figures 13 and 14, in the form of contour plots where each intersection of parameters is computed as the average of the error of the corresponding images and samples. In white is the level curve for the average of the FFT synthesized images for each combination of coverage percentage and metric, and shaded in white are the levels where the FFT method performs better than the GPR method. From these pictures it can be concluded that:

- There is an erratic behavior of the error when the parameter \(\lambda\) approaches 1 in order of magnitude, which will not be further studied in this section as this parameter is related to the noise in the measurements, and there is no noise being added neither on the original image nor in the Fourier space of it. This parameter is not set to 0 for practical reasons. As it can be seen in Equations (29) and (30), the results of the GPR require the inverse of the matrix \(K + \lambda I\), where \(K\) is positive semidefinite and symmetric. \(\lambda\) in this scenario avoids the indefiniteness of the matrix \(K + \lambda I\) and a larger \(\lambda\) means a smaller risk of loss of significance, with this, the idea is to use the largest possible \(\lambda\) that does not increase the error.

- In all of the cases there are several combinations of parameters where GPR outperforms FFT at least in average.

This poses a second part for this experiment, that is extracting the rest of the metrics with a fixed \(\lambda = 10^{-5}\) the results are shown in Figures 15 and 16. From this:

- With the MSE metric, the variation in performance of the GPR method is more significant than with WMSE, and the final relative difference from FFT is also higher than with WMSE.

- With respect to the coverage, it can be appreciated that the relative difference between GPR and FFT is slightly more in favor to the GPR in the lower coverage case, but it is not a significant improvement at this range, and supports the decision of not experimenting on several different coverages of the space.

- With the right kernel length, GPR method performs significantly better than FFT for any metric.

\footnote{Note that the axis for the parameter \(\lambda\) is logarithmic for these plots.}

\footnote{Note that the boxplot for the FFT method is replicated in each plot in order to avoid visual misconceptions.}
Figure 13: Heatmap representing the mean of the MSE for the different samples of each combination of hyper-parameters. Top: 15% of visibility, bottom: 30% of visibility. White section shows where GPR is outperformed by FFT.

Source: Own elaboration.
Figure 14: Heatmap representing the mean of the WMSE for the different samples of each combination of hyper-parameters. Top: 15% of visibility, bottom: 30% of visibility. White section shows where GPR is outperformed by FFT.

Source: Own elaboration.
Figure 15: Boxplots for image MSE, different values of $l$ (15% of visibility on top, 30% of visibility on bottom, dashed mean). FFT boxplot is replicated on both sides to avoid misconceptions.

Source: Own elaboration.
Figure 16: Boxplots for image WMSE, different values of $l$ (15% of visibility on top, 30% of visibility on bottom, dashed mean). FFT boxplot is replicated on both sides to avoid misconceptions.

Source: Own elaboration.
4.3 Optimization of the kernel length

For completeness, the third experiment consisted on focusing on the behavior of the kernel length hyper-parameter and finding an optimal value for the kind of images yielded by the generator with the chosen configuration.

For this experiment, a total of 410 images were generated. 5 visibility masks were sampled with a coverage of 15% of the space. For each of these, a referential FFT reconstruction was performed and afterwards a GPR reconstruction for each value of the hyper-parameter \( l \) in the set

\[
B_l = \{0.2 + 0.1 n\}, \quad n = 0, \ldots, 14.
\]  

The results for this experiment can be seen in Figure 17, where the solid green line corresponds to the mean of the error for the GPR synthesis over the images for each value of kernel length, while the shaded area covers one standard deviation from the mean. FFT in red for reference.

From this experiment the following conclusions can be extracted:

- The optimal GPR kernel length for both metrics is around 0.5, with a mean of less than a third WMSE of FFT and around a sixth MSE. It is appreciated that the variance is also significantly lower in both cases.

- From a qualitative analysis of the reconstructions, summarized in Figure 18, it can be seen that a lower kernel length spreads the noise towards the edge of the image while a higher kernel length seems to confine the information towards the center. This hints that the optimal kernel length confines the flux to an area roughly the same size as the original object.

4.4 Noisy samples

In the second experiment, the hyper-parameter \( \lambda \) was considered not relevant because of the absence of noise in the samples. This experiment studies the relevance of \( \lambda \) when a random normal noise is added to the samples.

The noise in this case is added in Fourier space, before the visibility sampling. In order to have control over the amount of noise, the notion of Signal to Noise Ratio (SNR) is used. Let the signal, denoted by \( S_{\text{RMS}} \), be the Root Mean Square (RMS) of the information in the
Figure 17: Mean and standard deviation (shaded) of the MSE (top) and WMSE (bottom) for the different values of $l$.

Source: Own elaboration.
Figure 18: Examples of GPR reconstruction with 15% of coverage and non-optimal kernel length, first: $l = 0.2$, second: $l = 1.1$.

Source: Own elaboration.
picture, i.e. the RMS of the pixel values, then the SNR is the following:

\[
\text{SNR} = \frac{S_{\text{RMS}}}{\varepsilon_{\text{RMS}}} = \sqrt{\frac{(\sum_{i=1}^{N} t_i^2) / N}{(\sum_{i=1}^{N} \varepsilon_i^2) / N}},
\]

(68)

where \( t_i \) represents the signal (actual value) of the pixel \( i \) and \( \varepsilon_i \) the noise in the same pixel. Note that \( S_{\text{RMS}} \) depends only on the function, while \( \varepsilon_{\text{RMS}} \) corresponds to the standard deviation of the noise. In order to achieve a desired SNR the noise must behave

\[
N \sim \mathcal{N}\left(0, \frac{S_{\text{RMS}}}{\text{SNR}} I\right).
\]

(69)

The set of values for the SNR in this experiment was

\[
B_{\text{SNR}} = \{2^n\}, \quad n = -2, \ldots, 5,
\]

(70)

d this is, from \( \text{SNR} = 0.25 \) which means the noise RMS takes up to 4 times the signal RMS, to 32, which makes a low noise scenario. The set of \( \lambda \) hyper-parameters was

\[
B_\lambda = \{2^n\}, \quad n = -4, \ldots, 7.
\]

(71)

In experiment, a total of 381 images were used. For each of these, 3 visibilities were sampled with a coverage of 15% of the space. For each visibility and SNR, a referential FFT reconstruction was performed and afterwards a GPR reconstruction for each value of the hyper-parameter \( \lambda \) in the set.

The results of this experiment can be seen as heatmaps in Figure 19, where both axes are in logarithmic scale because of the values chosen for SNR and \( \lambda \). It is fair to notice that in both cases, the lowest error achieved by FFT synthesis is not low enough to enter the GPR scale. Because of this, the FFT reference results are shown in a different chart in Figure 20.

From this experiment can be concluded that

- Even when it seems there is a relation between the hyper-parameter lambda and the SNR, it is not as evident as expected. Although further studying this relation could be highly relevant for the astronomical application, it is a more fundamental question that escapes the scope of this work.

- GPR shows dominance when it comes to error in the range of \( \lambda \) values studied, yet no optimality or near-optimality claims can be made here. This is also an interesting research line to address in the future.
Figure 19: Heatmap representing the mean of the MSE (top) and WMSE (bottom) for the different samples of each combination $\lambda$/SNR.

Source: Own elaboration.
4.5 **Lambda exploration in noisy samples**

This experiment comes as a correction to the range of \( \lambda \) values studied in the previous one. Here, 238 images were generated, each one with 3 different visibility samplings with a coverage of 15\% of Fourier space. In concordance with the previous experiment, the results are shown using the same kind of heatmaps in Figure 21.

It can be seen in both heatmaps that there is no further improvement while decreasing the value of this hyper-parameter. With these results, a single \( \lambda \) value in the constant region can be chosen to further study the behavior of both methods. This is shown in Figure 22.

From these results it can be concluded that

- As SNR decreases, FFT shows a sustained increase in the error in both cases, while GPR quickly saturates this increase and achieves to maintain the error, thus proving its robustness to noise.

- Even in the presence of noise, \( \lambda \) values up to 0.01 do not make a difference in the error.
Figure 21: Heatmap representing the mean of the MSE (top) and WMSE (bottom) for the different samples of each combination $\lambda$/SNR. Second iteration.
Source: Own elaboration.
Figure 22: Mean and standard deviation of the MSE (top) and WMSE (bottom) for FFT and GPR ($\lambda = 2^{-9}$) for different SNR values.

Source: Own elaboration.
4.6 Astronomical-like Visibilities

The last systematic experiment consists on simulating the rotation of the antennas with respect to the object under observation, to provide a more reasonable astronomical-like coverage of the uv-plane and with this, closer conditions to the ones in the focused application. In order to generate these masks the following algorithm was used:

- A random vector of 100 initial positions was generated in polar coordinates, this is, a random \( r \in (0, 1) \) and \( \theta \in [0, 2\pi) \) for each sample.

- An fixed increment of \( \pi/40 \) was defined for \( \theta \), while for each set of visibilities, a random increment \( \Delta r \in [0, 1/50] \) was generated.

- Each original sample is replicated with 14 successive increments, yielding a vector of 1500 visibilities in polar coordinates.

- Each of these final visibilities is placed upon the mask, activating the corresponding spot. Finally, all the activated pixels on a 3 pixel radius from the center are deactivated because each of those points represents a baseline or distance between antennas, which cannot be too close to zero. Some examples of the resulting sets from this algorithm can be seen in Figure 23.

Due to this last condition and also to superposition between visibilities it is fair to notice that the coverage at the end is not 15% but it is in a range between 9.74% and 13.18% with a mean close to 11.6%. This experiment was performed over 500 generated images, with 5 different sets of visibilities generated for each of these. The GPR reconstruction was performed with the already established hyper-parameter for this set of experiments \( \lambda = 10^{-3} \) and \( l = 0.5 \). An example of this process can be seen in Figure 24.

From the results of this experiment, shown in Figure 25, it can be concluded that:

- GPR keeps having a much lower error, nevertheless, there is no significant relative improvement from the uniformly random visibilities, when contrasting with Figures 15 and 16.

- The breach between the results from both metrics shrunk almost completely, with very similar results for MSE and WMSE.

4.7 GPR Synthesis on a real image

This qualitative experiment was designed in order to take a glance at the method potential beyond the specified functions. As described in the diagram in Figure 26, this time the orig-
Figure 23: Three different examples of visibility simulations. 
Source: Own elaboration.

Figure 24: Example of synthesis with astronomical-like visibility set. Top left: original image, top right: coverage of the uv-plane, bottom left: FFT reconstruction, bottom right: GPR reconstruction ($\lambda = 10^{-3}, l = 0.5$). 
Source: Own elaboration.
Figure 25: Boxplot for MSE and WMSE of astronomical-like visibility simulations (dashed mean).

Source: Own elaboration.

Figure 26: Diagram of the qualitative experiment on the real image.

Source: Own elaboration.
Figure 27: Picture of Coyoya the cat, reconstructed from sampling of 30% of Fourier space using $l = 0.1$ (top) and $l = 0.01$ (bottom).

Source: Own elaboration.
inal image is a real world picture, which is transformed numerically to Fourier space, to be sampled and then synthetized via GPR.

As according to the previous experiments the value of $\lambda$ makes no great difference beyond certain point, this value was set to $10^{-3}$, while different values for $l$ were tried. The reconstructions can be appreciated in Figure 27. Similar to what was observed in Figure 18, a higher value of $l$ focuses on the features towards the center of the image, while a lower value achieves an overall noisier image but with a more similar shape.
CHAPTER 5
CONCLUSIONS

5.1 On the objectives of the work

This work was based on four specific objectives focused on serving a single general objective. Regarding the specific objectives:

- A prototype of the GPR Image Synthesis method was successfully implemented and added to the appendix section of this document.

- Originally the idea was to propose an approach in one dimension, but given the practical applications it was decided to attempt a two dimensional approach. The vectorial nature of the formulation of this approach exposed on Section 2.5 makes this method able to be extended further in dimensionality.

- MSE fulfilled being a general purpose measuring criteria, as for some applications different from the one that motivated this work might be more significant. WMSE was better suited for the intended application. Given the kind of images used, the main difference between the results in both was due to the presence of noise around the edges of the reconstruction.

- Using the prototype and the metrics described, several experiments were performed against an already implemented version of the currently used FFT Image Synthesis method, measuring different aspects of interest over procedurally generated images.

Overall, through achieving these specific objectives, GPR Image Synthesis was successfully proposed and validated. Although the results over synthetic data are not conclusive of the performance of the method in real scenarios, the method shows promising results. As it can be seen in the results from Equation (51) and the code in Appendix 6.1, the proposed method achieves a closed-form expression for the inference that depends directly on the original samples, also no further processing is required to get the final result, as opposed to the current FFT approach and in concordance with the general objective of this work.

5.2 On the results of the method

Throughout the different experiments, GPR managed to outperform FFT in both metrics. As a summary of the most important experiment conclusions, in normal conditions, GPR achieved an MSE approximately six times better than FFT and a WMSE of approximately three times
better, along with a much smaller variance of error across experiments. GPR also shows robustness against noise by having a lower slope in Error versus SNR charts.

Regarding the astronomical-like visibilities, the results were quite similar to the same experiment with uniform visibilities. It was thought initially that GPR might have even better results in these conditions because in the gridding process with unevenly distributed samples, the grid would have some nodes with no information close by and some others with a concentration of information, also affecting this convolution based algorithm.

However, GPR’s further advantage cannot be ruled out yet. There is still much research to be done in this area given that some factors, while relevant for the motivation domain of this work, fell out of its scope. For instance, even when hyper-parameter tuning could raise the MSE, it could at the same time affect the number of artifacts that would appear on the picture, or have other qualitative effects desirable for the astronomical domain.

5.3 On the hyper-parameters

The first hyper-parameter of the model, named $\lambda$ (Equation (24)), is native to the Gaussian Process Regression. It was thought initially for this parameter to have a direct relation to the noise in the samples, however, at least with the type of images used in this work and the range of SNR used, it was found to have no impact in the GPR mean if sufficiently low. It is very likely however to have a significant impact in the GPR variance, which was not studied in here.

This makes $\lambda$ necessary only for numerical reasons. Since the kernel matrix $K$ is known to be positive semidefinite for the conditions on the kernel function, some of its eigenvalues might be zero, preventing the matrix from having an inverse. $\lambda > 0$ ensures that the eigenvalues of $K + \lambda I$ are strictly positive and thus the inverse matrix $(K + \lambda I)^{-1}$ exists.

The second hyper-parameter of the model, named $l$ or kernel length (Equation (44)), comes from the chosen kernel function. As stated before, the optimal length for the kernel is different for different kinds of pictures, which can be seen in how different were the range of values used in most of the experiments to the ones used for the reconstruction in Figure 27.

This can be explained from the physical meaning of the kernel length. As it can be seen in Equation (44), this parameter corresponds to the standard deviation of the Gaussian kernel function, which basically takes as input the distance between two points in Fourier space. As this space is 50 units long and the representation takes 100 pixels, each pixel takes 0.5 units in length, leaving the 3σ radius at 1.5 pixels. In other words, for the inference of any point in the Fourier surface only the samples that are less than 1.5 pixels away will have a real impact in the value.
5.4 On the time performance of the methods

This work was not focused on competing against the current method time-wise, as one of the main reasons why the FFT is used in image synthesis is actually time performance. A plain two dimensional DFT based algorithm would have a temporal complexity of $O(n^2)$ with $n$ being the number of pixels in the image as it would consist on a linear transform using the DFT Matrix as shown in Equation (12), while the FFT has a temporal complexity of $O(n \log n)$ [Sauer, 2012].

In the method proposed on this work, the FT is analytically applied, but the inference on the image requires the product of an inverse Matrix with a vector, which is performed by solving a system of linear equations with a complexity of $O(m^3)$, with $m$ the number of measurements obtained. Taking into account the rest of the inference, which consist on a vector product for each pixel, the total complexity of the method would be $O(m^3 + mn)$.

It is fair to mention that the prototype takes a couple of orders of magnitude more than the FFT implementation, but computing power has also increased that much since the first time this FFT-based algorithms were implemented, and also as a prototype it lacks the decades of depuration the FFT implementation has.

5.5 Future work

5.5.1 Time Performance

As mentioned before, time performance is the main weakness of GPR synthesis, nevertheless, there is no evident setback in the paralelization of the algorithm, as the inference involves mainly algebraic operations. One possible branch of improvement is to implement the algorithm in a parallel paradigm to run in GPU or HPC, thus taking full advantage on the computing power of current setups.

5.5.2 Variance

Another advantage of the GPR is that defines a probability distribution over the values. It does not only return the inferred surface or mean, but also the variance for each prediction. Throughout this work, this feature has not been used, but it could be a useful indicator for the certainty of each prediction across a surface. On-going research points towards this variance also having a closed-form using Gaussian kernels.

---

12 or $O(k^4)$ in a $k \times k$ image.
13 or $O(k^2 \log k)$ in a $k \times k$ image.
5.5.3 Tuning of the Algorithm

Regarding the astronomical application, this method can be further specialized for it in several ways, that include:

- Selecting a different kernel that represents better the influence that a baseline measurement should have in the others.

- Tuning of the hyper-parameters of the method and selected kernel to fit the real astronomical data.

- Defining a different a priori distribution of the expected image, as it is supported by GPR but was not included in this work.

- General further research on the topics that fell out of the scope of this work, as it was mentioned several times.
REFERENCES


import numpy as np
from numpy import fft
from scipy import spatial as spt

def gpr_synthesize_image(surface, visibility, imaging_size, k_length, lambd):
    # Sampling of the visibilities, from the given mask.
    image_data = surface[visibility]
    # Getting visibility points in Fourier's space domain.
    domain_base = fft.fftshift(fft.fftfreq(imaging_size, 2/imaging_size))
    domU, domV = np.meshgrid(domain_base, domain_base)
    sampU = domU[visibility]
    sampV = domV[visibility]

    # Computation of (K + lambda * I).
    C = np.exp((
        - spt.distance_matrix(np.matrix(sampU).T,
                                np.matrix(sampU).T) ** 2
        - spt.distance_matrix(np.matrix(sampV).T,
                                np.matrix(sampV).T) ** 2
    )/(2*k_length**2)) + lambd * np.eye(image_data.shape[0])

    # Computation of the product between the inverse of 
    # (K + lambda * I) and the vector of visibilities v_0.
    psi = np.linalg.solve(C, image_data)

    # Computation of the rest of the expression.
    domain_base = np.linspace(-1, 1, IM_SIZE)
    _, _, sampU = np.meshgrid(domain_base, domain_base, sampU)
    ex_domX, ex_domY, sampV = np.meshgrid(domain_base, domain_base, sampV)
    gs = np.exp(2j * np.pi * (ex_domX*sampU + ex_domY*sampV))

    # Imaging domain and final result.
    return np.abs(2 * np.pi * k_length**2 *
                   np.exp(-2*np.pi**2*k_length**2*(domX**2+domY**2)) *
                   gs.dot(psi))
This function takes as parameters the Fourier surface of the image, which is a two dimensional square Numpy array, a mask of visibilities, which is a boolean array with the same shape of the surface, the desired size of the output image (size of the side of the square) and the hyper-parameters of the GPR model.