A Gaussian Process Regression Framework for Spatial Error Concealment with Adaptive Kernels

Hadi Asheri, Hamid R. Rabiee, Nima Pourdamghani, Mohammad H. Rohban
AICTC Research Center, Department of Computer Engineering, Sharif University of Technology
rabiee@sharif.edu

Abstract

We have developed a Gaussian Process Regression method with adaptive kernels for concealment of the missing macro-blocks of block-based video compression schemes in a packet video system. Despite promising results, the proposed algorithm introduces a solid framework for further improvements. In this paper, the problem of estimating lost macro-blocks will be solved by estimating the proper covariance function of the Gaussian process defined over a region around the missing macro-blocks (i.e. its kernel function). In order to preserve block edges, the kernel is constructed adaptively by using the local edge related information. Moreover, we can achieve more improvements by local estimation of the kernel parameters. While restoring the prominent edges of the missing macro-blocks, the proposed method produces perceptually smooth concealed frames. Objective and subjective evaluations verify the effectiveness of the proposed method.

1. Introduction

The video coding standards use block prediction and entropy coding in order to achieve more efficient compression [1]. Transmission of such video streams over error-prone channels is susceptible to packet loss due to congestion, fluctuating channel errors and error propagation due to the compression. Spatial error concealment (SEC) is the process of recovering lost macro-blocks (MB) by employing the correlation between the lost MB and its correctly received neighboring MBs [2]. Previous works on SEC include methods that estimate missing blocks in transform domain [3]. Hybrid methods [4, 5, 6] employ the DCT coefficients to produce some smoothing constrains in order to estimate the missing block [4, 5] or to generate some interpolation matrices [6]. Other methods employ spatial domain information for estimation. Bilinear interpolation (BI) [7] is a simple and effective method that was employed in H.264/AVC standard [8]. The above algorithms, more or less ignore the edge information, and hence produce results that are smooth approximations of the missing blocks. On the other hand, directional SEC methods use edge related information [9, 10, 11, 12, 13]. Authors in [9] perform a projection onto a convex set after estimation of the most likely edge orientation. The concept of sequential error concealment is considered in [10]. An adaptive spatial pixel interpolation is proposed in [11]. The method proposed in [12] uses spatial direction vectors (SDVs) extracted from edge information, it then applies SDVs to adaptively interpolate missing pixels. Authors in [13] have proposed an approach to integrate the state of the art spatial error concealment methods. Avoiding the production of false edges is as important as preserving existing edges. This idea is addressed in [14]. Best Neighborhood Matching is proposed in [15]. Some statistical models [16] have also been applied to SEC.

The proposed method in [17] has touched the concept of Gaussian Process Regression (GPR) but it mostly focuses on sequential concealment and doesn’t investigate on the problem of kernel construction which is the main concern of our work. In this paper, we attempt to interpolate missing pixels by employing Gaussian Process Regression [18], which provides a non-parametric Bayesian approach to regression problem. Having a great foundation in statistics and machine learning, Gaussian Process modeling is a general and rich framework that is related to a number of other popular models such as Spline models, Support Vector Machines, Least-Square methods, Relevance Vector Machines and Wiener filters [18].

The rest of this paper is organized as follows. Section 2 proposes a novel GPR approach to Spatial Error Concealment problem, and uses kernel parameter tuning approach to achieve efficient locally adaptive kernels. Section 3 describes our experimental implementation and discusses the results. Finally section 4 addresses
limitations of our work and some possible future investigations.

2. Gaussian Process Regression

In this section, we present the proposed GPR approach to spatial error concealment. After formulating the SEC as a GPR problem, we explain how to utilize edge related information to construct the kernel function. At the end, the marginal likelihood maximization has been introduced as a typical but suboptimal and time consuming method to kernel parameter estimation.

2.1 Gaussian Processes

Over the last decade, kernel machines have become rather popular in the context of machine learning. During this period, much work has been done concerning the application of Gaussian process models in this field. Gaussian processes provide a practical, analytical and statistical framework for the problem of kernel machine learning. A Gaussian process is a generalization of Gaussian distributions to functions. Whereas Gaussian distributions describe random scalars or vectors, Gaussian processes govern the properties of functions. Supervised learning problem which can be thought of as learning a function given some sample points, can be directly mapped to the Gaussian process framework. By employing relevant covariance functions, Gaussian processes will be equivalent to many well known models such as large-scale neural networks, spline models, support vector machines and so on. A Gaussian process is completely specified by its mean and positive definite covariance function. We will write the Gaussian process as:

\[ f(p) \sim GP(m(p), k(p, p')) \]  

Where \( m(p) \) denotes its mean function for each pixel \( p = (x, y) \) and \( k(p, p') \) is the covariance function of two pixels \( p = (x, y) \) and \( p' = (x', y') \). For simplicity, we assume that \( m(p) = 0 \) thus the problem will be reduced to finding the covariance function. This covariance function can be approximated by a variety of different kernel functions. Different kernels will result in different models. For example, if we use the simple Euclidian distance (i.e. the inner product) as the kernel, one can show that a linear model will be obtained. Other kernels will result in more sophisticated models. Letting \( P \) denote neighboring correctly received pixels and \( f \) be their corresponding intensity values, we wish to estimate the intensity values \( f_s \) for pixels \( P_s \) of the missing block. As we have assumed the joint pdf of \( f \) and \( f_s \) have the form below:

\[
\begin{bmatrix}
  f \\
  f_s
\end{bmatrix} \sim N\left(0, \begin{bmatrix}
  K + \sigma_n^2 I & K^T \\
  K_s & K_{ss}
\end{bmatrix}\right)
\]  

(2)

Where \( \sigma_n^2 \) is the noise variance, \( I \) is the Identity matrix, \( K_s \) is a matrix denoting the kernel function evaluated at all pairs of pixels in \( P_s \) and \( P \) respectively and a similar definition holds for \( K \) and \( K_{ss} \). Knowing this pdf we can find the conditional distribution of \( f_s \) given \( P, P_f \) and \( f \):

\[
f_s|P, P_f, f \sim N(K_s(K + \sigma_n^2 I)^{-1} f, K_{ss} - K_s(K + \sigma_n^2 I)^{-1} K_s^T)
\]

(3)

In order to minimize the mean squared error, the best estimation for \( f_s \) will be the mean of the above normal distribution.

\[
\hat{f}_{MSE} = K_s(K + \sigma_n^2 I)^{-1} f
\]

(4)

It can be shown that the Gaussian process approach can be represented as the linear combination of \( n \) kernel functions

\[
f_s = \sum^n_{i=1} \alpha_i k(x_s, x_i)
\]

(5)

where \( \alpha \) is the coefficient vector with optimal value \( \alpha = (K + \sigma_n^2 I)^{-1} f \).

2.2 Kernel Construction

Although images are generally non-stationary, they can be locally modeled by stationary Gaussian Processes. As discussed in [18], for covariance functions \( k(x, x') \), there are various kernels such as linear, \( \gamma \)-exponential, rational quadratic, Matern and piecewise polynomial. We have chosen the \( \gamma \)-exponential covariance function \( k \):

\[
k(x, x') = \exp\left(\frac{-|x - x'|}{l}\right) \quad 0 < \gamma \leq 2
\]

(6)

where \( r = |x - x'| \), \( l \) is the scaling factor and \( \gamma \) is the gamma parameter. For each missing block we consider its three pixel wide boundary neighborhood in horizontal and vertical directions. We then find the dominant edge in the specified neighborhood by a gradient operator (the sobel mask). Let \( d \) be the Euclidian distance between two pixels \( x \) and \( x' \) and \( r \) denote the absolute difference of their Euclidian distances from the dominant edge. The kernel is defined as

\[
k(x, x') = \exp\left(\frac{d}{l_d}\right) \exp\left(\frac{r}{l_r}\right)
\]

(7)

\[\text{4542}\]
If there is no strong dominant edge in the neighboring area of the missing block, our kernel function will be as follows:

$$k(x, x') = \exp \left( \frac{d}{l_d} \gamma \right)$$ (8)

### 2.3 Kernel Parameter Tuning

Here, the main concern is to approximate the kernel functions efficiently. Marginal likelihood maximization would be a typical way to estimate kernel parameter in the case of parametric kernels. The marginal likelihood is the integral of the likelihood as follow

$$p(y|X, \theta) = \int p(y|f, X)p(f|X)df.$$ 

Under the Gaussian Process model there is a Gaussian prior, $f|X \sim N(0, K)$, or

$$L(\theta) = \log p(y|X, \theta) = -\frac{1}{2}f^T(K + \sigma_n^2 I)^{-1}f - \frac{1}{2}\log|K + \sigma_n^2 I| - \frac{n}{2}\log 2\pi$$ (9)

In order to maximize marginal likelihood, we employ the gradient descent algorithm. The gradient of marginal likelihood will be as follows:

$$\frac{\partial}{\partial \theta_j} \log p(y|X, \theta) = \frac{1}{2} \text{tr} \left( (\alpha\alpha^T - K^{-1}) \frac{\partial K}{\partial \theta_j} \right)$$ (10)

where $\alpha = K^{-1}y$.

### 3. Simulation Results

Marginal likelihood maximization did not prove itself as a satisfactory measure for parameter estimation in our experiments. Due to its computational complexity we have set the parameters by experimental evidence. We have set the parameter $\gamma$ to 1.8, $l_r$ to 2 and $l_d$ to four times the block size. The performance of the proposed algorithm is tested on various video sequences and images with the size of corrupted blocks being $16 \times 16$.

Table 1 contains the results of our experiments for error patterns as in Fig. 1(b, g). We have used Peak Signal to Noise Ratio (PSNR) as a comparison measure and compared our method with the well-known Bilinear interpolation [7]. For each video sequence the PSNR value have been averaged over frames 2, 12, ..., 92. As illustrated in Table 1, the proposed method has obtained an improvement of about 1.3~3.5 dBs over the Bilinear method (BI). Moreover, we have compared the proposed framework with the state of the art FDI method [12]. The results shown in Fig. 1 support our claim that this framework is superior to BI and comparable to FDI methods.

Fig. 1 shows the subjective comparison results of the two algorithms. From these figures we can observe that bilinear algorithm (Fig. 1(c, h)) causes blurring effects around the edges of the recovered areas. Fig. 1(d, i) depicts the results of the FDI method and Fig. 1(e, j) illustrates those of the proposed method. As the results show, missing areas have been reconstructed successfully by the proposed method without the edge blurring effect. As it can be seen, our method subjectively performs far better than BI and is comparable or better than the FDI method.

### 4. Conclusion

Our work was the first step in using Gaussian process modeling in video error concealment. It depicts a framework which seems to be effective in image and video processing. Although Gaussian processes have recently been applied to practical problems in machine learning, they have a rich theoretical background behind them. This has caused an appealing future for using them in this context. As this work illustrates, by utilizing a simple kernel we have gained a substantial improvement in both subjective and objective quantities of the reconstructed images. Future works might contain examining other kernels suitable for error concealment and an analytical view to different kernels which might be used. Also use of kernels might be extended to temporal error concealment and motion vector estimation.
Figure 1: Subjective comparison of the algorithms with blocks of size 16 × 16.

References