To appear in Mathematics of Expressive Image Synthesis II (Springer 2015).

# Superresolution from Principal Component Models by RKHS Sampling

J.P. Lewis Ken Anjyo Taehyun Rhee

**Abstract**

Principal component analysis (PCA) involves a signal that is sampled at some arbitrary but fixed and countable set of locations. Radial Basis Function (RBF) regression interpolates a-priori known data to arbitrary locations as a weighted sum of a (radial) kernel function centered at the data points. In recent work we showed that if the RBF kernel is equated to the covariance, RBF and Gaussian Process (GP) models perform a similar computation, differing in what information is assumed known in advance, and what is known at runtime. Building on the RBF-GP equivalence, we show that if the data covariance is known (or can be estimated), an RBF-inspired regression can provide data-driven "superresolution" interpolation of given data. This procedure can alternately be interpreted as a superresolution extension of eigenvector (principal component) data models, as signal sampling (function evaluation) in a discrete reproducing kernel Hilbert space (RKHS) generated by the data covariance, or as an elementary Gaussian process model in which the observations have a low-rank representation.

## 1 Introduction

In this paper we demonstrate a data-driven superresolution procedure. The procedure can alternately be interpreted as a superresolution extension of principal component analysis (PCA) data models, or as function evaluation (signal sampling) in a discrete reproducing kernel Hilbert space (RKHS) generated by the data covariance.

**Terminology.** The paper informally equates a radial basis function $G(\Delta)$ to a covariance. In so doing we consider covariances $C(x, y) = C(\|x - y\|) = C(\Delta)$ corresponding to stationary processes. Throughout the paper the mean of the process or data is assumed to be zero without loss of generality. We further use the covariance as the kernel in an RKHS. The reader should also distinguish continuous functions and the matrices generated by evaluating the continuous function at discrete locations. Discrete variables are indicated in bold font. $M(d, n, \mathbb{R})$ denotes the set of real matrices of size $d \times n$. The number of data points is denoted by $n$, $d$ is the dimensionality of a single data point in the case of PCA data, and $m$ is the number of eigenvectors retained in a PCA model. In

order to suggest a relationship, related quantities are sometimes denoted with the same symbol. The usage is explained and is unique within each section.

**Principal Component Analysis.** PCA represents data in a linear subspace obtained from the eigenvectors of the covariance matrix, estimated as $\mathbf{C} \approx \frac{1}{n}\mathbf{F}\mathbf{F}^T$ where $\mathbf{F} \in M(d, n, \mathbb{R})$ is a matrix containing the data points $\mathbf{f}_k \in \mathbb{R}^d$ in its columns. We refer to a PCA "model" as

$$\mathbf{f} = \mathbf{U}\mathbf{c} + \mathbf{m} \tag{1}$$

where $\mathbf{f}$ is a vector representing the signal being modeled, $\mathbf{U} \in M(d, m, \mathbb{R})$ are eigenvectors of the covariance matrix $\mathbf{C}$ corresponding to the $m$ largest eigenvalues ($m \leq d$), $\mathbf{m} \in \mathbb{R}^d$ is the mean of the data points, and $\mathbf{c}$ are PCA "coefficients". In this model the data covariance

$$\mathbf{C} = \mathbb{E}\left[(\mathbf{f} - \mathbf{m})(\mathbf{f} - \mathbf{m})^T\right] = \mathbb{E}\left[\mathbf{U}\mathbf{c}\mathbf{c}^T\mathbf{U}^T\right]$$

is replaced with the low rank approximation

$$\mathbf{C} \approx \mathbf{U}\mathbf{\Lambda}_{1:m}\mathbf{U}^T$$

where $\mathbf{\Lambda}_{1:m} \in M(m, m, \mathbb{R})$ denotes the diagonal matrix formed from the largest eigenvalues.

Strictly speaking PCA is a valid model only in the case where the data is jointly Gaussian. Nevertheless, this approximate model is adequate in some applications. For example, a jointly Gaussian model of face proportions has been widely employed in computer vision [6] and even psychological theory [15]. While PCA is not considered a generative model, it is easy to synthesize new data having the same covariance as the observed data $\mathbf{f}_k$ by picking random coefficients $\mathbf{c}$ according to $c_k^2 \sim \lambda_k$.

To simplify notation, in the remainder of the paper $\mathbf{\Lambda}_{1:m}$ will be denoted simply as $\mathbf{\Lambda}$. Also, the covariance matrix $\mathbf{C}$ should be understood as representing either the full rank covariance matrix or a low-rank approximation thereof. Most of the calculations are unchanged, although $\mathbf{C}^{-1}$ and $\mathbf{\Lambda}^{-1}$ should be understood as pseudoinverses in the low-rank interpretation.

**RBF regression.** RBF regression at a location $\mathbf{p}$ has the form

$$\hat{f}(\mathbf{p}) = \sum_{k=1}^{n} a_k G(\|\mathbf{p} - \mathbf{p}_k\|)$$

where $G()$ is a radial function situated at the $n$ training data locations $\mathbf{p}_k$. Assembling the data values to be interpolated in a vector $\mathbf{f}$, the weights can be obtained from the linear system

$$\mathbf{G}\mathbf{a} = \mathbf{f}$$

where $\mathbf{G}$ is the $n \times n$ matrix consisting of the radial basis function evaluated at all pairs of data locations, $\mathbf{G}_{r,c} = G(\|\mathbf{p}_r - \mathbf{p}_c\|)$. This describes the case

of regression with a positive definite radial kernel $G$. Other RBF kernels are conditionally positive definite and require solving a block matrix system [3]. In matrix-vector notation the regression can be written

$$\hat{f}(\mathbf{p}) = \mathbf{r}^T \mathbf{G}^{-1} \mathbf{f} \tag{2}$$

where $\mathbf{r} \equiv \mathbf{r_p}$ is the vector of the kernel evaluated at the distance between the evaluation location $\mathbf{p}$ and all the training locations $\mathbf{p}_k$, i.e.,

$$\mathbf{r} = [G(\|\mathbf{p} - \mathbf{p}_1\|), G(\|\mathbf{p} - \mathbf{p}_2\|), G(\|\mathbf{p} - \mathbf{p}_3\|), \cdots, G(\|\mathbf{p} - \mathbf{p}_n\|)] \; .$$

Note that the position $\mathbf{p}$ is arbitrary.

**Gaussian process regression.** A discrete Wiener filter estimates the signal $\hat{f}(\mathbf{p})$ at a location $\mathbf{p}$ as a linear weighted sum of the data $\mathbf{f}$ at previous or surrounding locations,

$$\hat{f}(\mathbf{p}) = \mathbf{w}^T \mathbf{f} \, . \tag{3}$$

The orthogonality principle states that the error of the optimal estimator is orthogonal in expectation to the data:

$$\mathbb{E}\left[\left(f(\mathbf{p}) - \mathbf{w}^T \mathbf{f}\right) \mathbf{f}^T\right] = 0$$

or equivalently

$$\mathbb{E}\left[f(\mathbf{p}) \mathbf{f}^T\right] = \mathbf{w} \mathbb{E}\left[\mathbf{f}\mathbf{f}^T\right] \; .$$

$\mathbb{E}\left[\mathbf{f}\mathbf{f}^T\right] = \mathbf{C}$ is the covariance of a zero-mean stationary process and $\mathbb{E}\left[f(\mathbf{p})\mathbf{f}^T\right] = \mathbf{r}^T$ is the cross covariance between the signal at the location $\mathbf{p}$ and at the data points $f(\mathbf{p}_k) \equiv f_k$ for $k = 1, \cdots, n$. This gives a linear system $\mathbf{C}\mathbf{w} = \mathbf{r}$ that can be used to solve for the weight vector $\mathbf{w}$. Substituting in Eq. (3) gives the estimator

$$\hat{f}(\mathbf{p}) = \mathbf{r}^T \mathbf{C}^{-1} \mathbf{f} \, . \tag{4}$$

The variance of the estimator is

$$\begin{aligned} \mathbb{E}\left[\left(f - \mathbf{w}^T \mathbf{f}\right)^2\right] &= \sigma^2 + \mathbb{E}\left[-2 f \mathbf{w}^T \mathbf{f} + \mathbf{w}^T \mathbf{f}\, \mathbf{f}^T \mathbf{w}\right] \\ &= \sigma^2 - 2\, \mathbf{w}^T \mathbb{E}\left[f \mathbf{f}\right] + \mathbf{w}^T \mathbf{C} \mathbf{w} \end{aligned}$$

where $\sigma^2$ is the variance of the (stationary) process, and $f \equiv f(\mathbf{p})$ is written for brevity. Substituting the cross-covariance $\mathbf{r}$ and the weights $\mathbf{w} = \mathbf{C}^{-1}\mathbf{r}$, we have

$$\begin{aligned} &= \sigma^2 - 2 \mathbf{w}^T \mathbb{E}\left[f \mathbf{f}\right] + \mathbf{r}^T \mathbf{C}^{-1}\, \mathbf{C}\, \mathbf{C}^{-1} \mathbf{r} \\ &= \sigma^2 - \mathbf{r}^T \mathbf{C}^{-1} \mathbf{r} \, . \end{aligned}$$

Eq. (4) and this expression for the variance have the same form as the mean and variance in Gaussian process (GP) regression [12, 2.19].

In recent work [2] we observed that if the radial basis kernel function is equated to the covariance, RBF regression has the same computational form

as Gaussian Process regression, as can be seen from Eqs. (2) and (4). This correspondence requires simultaneous generalization of the RBF kernel to non-radial form and a restriction to valid (i.e. positive-(semi)definite) covariances.

In the remainder of this paper we first describe the superresolution procedure. We show that the procedure has the representer of evaluation and reproducing kernel properties of an RKHS, albeit in discrete rather than continuous form. Finally, we show a computational experiment using the superresolution procedure.

## 2 Superresolution

We start from the matrix-vector representation of RBF regression, (2). We will equate $G$ to a covariance function $C$, which is a generalization to non-radially symmetric kernels. "Oriented" flavors of RBF regression have previously appeared in the literature [7, 5]. Since the matrix $\mathbf{G}$ is symmetric, it has a low-rank approximation $\mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$. Substituting this eigenexpansion, the regression takes the form

$$\hat{f}(\mathbf{p}) = \mathbf{r}^T\mathbf{C}^{-1}\mathbf{f} = \mathbf{r}^T\mathbf{U}\mathbf{\Lambda}^{-1}\mathbf{U}^T\mathbf{f} \tag{5}$$

Denoting $\mathbf{c} \equiv \mathbf{\Lambda}^{-1}\mathbf{U}^T\mathbf{f}$, Eq. (5) can be interpreted as incorporating the results of a PCA-like representation $\mathbf{U}\mathbf{c}$ involving the data covariance eigenvectors $\mathbf{U}$ (we again assume the data mean is zero):

$$\begin{aligned} \hat{f}(\mathbf{p}) &= \mathbf{r}^T\mathbf{U}\left(\mathbf{\Lambda}^{-1}\mathbf{U}^T\mathbf{f}\right) \\ &= \mathbf{r}^T\mathbf{U}\mathbf{c} \end{aligned} \tag{6}$$

The discrete principal component model Eq. (1) involves data that is sampled at some arbitrary but fixed and finite set of locations. Eq. (6) extends this by premultiplying with $\mathbf{r}$. Recall that $\mathbf{r}$ can be interpreted as a vector of cross-covariances between the location $\mathbf{p}$ at which the the regression is evaluated and the locations of the data points $\mathbf{p}_k$,

$$\mathbf{r} = C(\mathbf{p}, \cdot) = [C(\|\mathbf{p} - \mathbf{p}_1\|), C(\|\mathbf{p} - \mathbf{p}_2\|), C(\|\mathbf{p} - \mathbf{p}_3\|), \cdots, C(\|\mathbf{p} - \mathbf{p}_n\|)] \tag{7}$$

If the covariance function $C(\mathbf{\Delta})$ is known for all offsets $\mathbf{\Delta}$ then the data estimate (6) can be evaluated at *arbitrary* locations, effectively providing a covariance-driven superresolution.

We next sketch that function generated with the superresolution estimate (6) in fact has covariance $C$. In this section let $\mathbf{r}_i$ denote the cross-covariance (7) evaluated at the location $\mathbf{p}_i$ (rather than at an arbitrary location). The covariance of the estimated signal is:

$$\begin{aligned} \mathbb{E}\left[f(\mathbf{p}_i)f(\mathbf{p}_j)\right] &= \mathbb{E}\left[\mathbf{r}_i^T\mathbf{C}^{-1}\mathbf{f}\,\mathbf{f}^T\mathbf{C}^{-1}\mathbf{r}_j\right] \\ &= \mathbf{r}_i^T\mathbf{C}^{-1}\,\mathbf{C}\,\mathbf{C}^{-1}\mathbf{r}_j \\ &= \mathbf{r}_i^T\mathbf{C}^{-1}\mathbf{r}_j \\ &= \mathbf{C}_{i,j} \end{aligned}$$

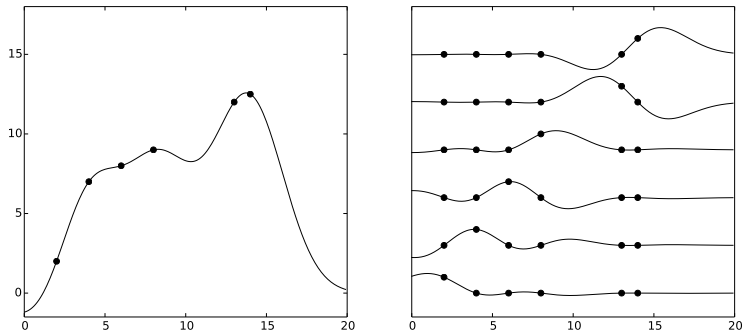


Figure 1: Left: Radial basis interpolation of six points using a Gaussian kernel. Right: equivalent kernels for each of the data points, offset vertically for visibility.

(For the last step note that $\mathbf{r}_i$ is the $i$th row of $\mathbf{C}$, hence multiplying by $\mathbf{C}^{-1}$ gives $\mathbf{e}_i$, the indicator vector with 1 in the $i$th location, otherwise zero).

# 3 Equivalent kernel

Note that Eq. (4) is more commonly interpreted in a different way. In this interpretation

$$\begin{aligned}\hat{f}(\mathbf{p}) &= \left(\mathbf{r}^T\mathbf{C}^{-1}\right)\mathbf{f} \\ &= \mathbf{b}(x)^T\mathbf{f}\,.\end{aligned}$$

Here $\mathbf{b}(x) = \mathbf{C}^{-1}\mathbf{r}$ is the vector of *equivalent kernels* [9] evaluated at location $x$. The individual kernels $\mathbf{b}_k(x)$ are one at the location $x_k$ of the $k$th data point, and zero at the locations of the other data points (Fig. 1).

# 4 Interpretation as evaluation in an RKHS

In this section we show that Eq. (5) resembles function evaluation in an RKHS corresponding to the covariance $\mathbf{C}$.

### Representer of evaluation

The kernel in an RKHS acts similarly to a delta function, providing function evaluation at a particular location under the inner product:

$$\langle f, K(\cdot, x)\rangle_C = f(x) \tag{8}$$

In our discrete case we propose that

$$\mathbf{f}^T\mathbf{C}^{-1}\mathbf{r}_x = f(j) \tag{9}$$

with $\mathbf{r}_x$ defined as the vector of $K(\cdot, x)$ evaluated at the data points. In the case of where $x$ is one of the data locations $\mathbf{p}_j$,

$$\mathbf{r}_x = \mathbf{C}\mathbf{e}_j$$

($\mathbf{e}_m$ is the indicator vector). Then, substituting $\mathbf{r}_x$, we have

$$\mathbf{f}^T\mathbf{C}^{-1}\mathbf{C}\mathbf{e}_j = f(j)\,.$$

## Reproducing property

Similarly,

$$\begin{aligned}\langle K(\cdot, x), K(\cdot, y)\rangle &= K(x, y)\\ \approx \left\langle \mathbf{e}_x^T\mathbf{C}, \mathbf{C}\mathbf{e}_y\right\rangle_C &= \mathbf{e}_x^T\mathbf{C}\,\mathbf{C}^{-1}\,\mathbf{C}\mathbf{e}_y = \mathbf{e}_x^T\mathbf{C}\mathbf{e}_y = \mathbf{C}_{x,y}\,.\end{aligned}$$

## Inner Product

The analogy between (8) and (9) requires an inner product of the form

$$\langle \mathbf{v}_1, \mathbf{v}_2\rangle_C = \mathbf{v}_1^T\mathbf{C}^{-1}\mathbf{v}_2\,.$$

This can be motivated as follows: in the continuous case, the reproducing kernel has the (Mercer) expansion:

$$\mathbf{C}(x, y) = \sum_{k=1}^{\infty}\lambda_k\mathbf{u}_k(x)\mathbf{u}_k(y)$$

A signal is also representable in terms of the eigenvectors of the kernel:

$$f(x) = \sum_{k=1}^{\infty}c_k\mathbf{u}_k(x)\,.$$

The inner product is defined to be

$$\left\langle \sum_{k=1}^{\infty}c_k\mathbf{u}_k(x), \sum_{k=1}^{\infty}d_k\mathbf{u}_k(x)\right\rangle_C = \sum_{k=1}^{\infty}\frac{c_k d_k}{\lambda_k}$$

giving the norm $\sqrt{\sum_{k=1}^{\infty}\frac{c_k^2}{\lambda_k}}$.

In practice in the discrete case the summation is finite. The squared norm can be written in matrix-vector form

$$\sum_{k=1}^{m}\frac{c_k^2}{\lambda_k} = \mathbf{c}^T\mathbf{\Lambda}^{-1}\mathbf{c}$$

and re-expressed in terms of the kernel as

$$\sum_{k=1}^{m}\frac{c_k^2}{\lambda_k} = \mathbf{f}^T\mathbf{U}\mathbf{\Lambda}^{-1}\mathbf{U}^T\mathbf{f} = \mathbf{f}^T\mathbf{C}^{-1}\mathbf{f}\,.$$

We will take the kernel as the covariance matrix, and for simplicity consider only the case of positive definite covariances. In this case all the eigenvalues are positive real and the norm is valid.

# 5 Computational Experiment

The superresolution estimate (6) requires supplying or estimating a covariance matrix. In some cases the form of the covariance matrix is known or assumed. For example, the use of the Discrete Cosine Transform (DCT) in image compression is motivated by the choice of $\mathbf{C}_{r,c} = \rho^{|r-c|}, \rho \approx 0.9$ as a generic covariance for images [11]. Estimating a nonparametric covariance matrix from data requires care if the number of data points is not sufficient to reliably estimate the $n(n+1)/2$ parameters of the covariance matrix, and is a subject of continuing research [10]. Although the covariance must be estimated, even an approximate covariance might result in better interpolation than would be obtained using generic spline interpolation.

There are known algorithms for generating multivariate normal random vectors with a specified covariance [8]. One simple algorithm is obtained by considering the covariance of a linearly transformed random vector $\mathbf{x} = \mathbf{T}\mathbf{n}$ where $\mathbf{n}$ is an uncorrelated normal random vector (i.e. having an identity covariance matrix):

$$\mathbf{C}_x = \mathbb{E}\left[\mathbf{T}\mathbf{n}\mathbf{n}^T\mathbf{T}^T\right] = \mathbf{T}\mathbf{T}^T .$$

Thus, if the transform $\mathbf{T}$ is a "square root" of a desired covariance matrix $\mathbf{C} = \mathbf{T}\mathbf{T}^T$, then $\mathbf{x}$ will have covariance $\mathbf{C}$ (the decomposition is not unique). $\mathbf{T}$ can be obtained using the Cholesky decomposition if the covariance is non singular. This algorithm is limited to generating vectors $\mathbf{x}$ with a fixed number of samples at offsets corresponding to those that generated the covariance matrix, whereas our procedure can generate the signal at arbitrary locations.

In Figs. 2-4 we simply select a covariance for the purpose of illustration, whereas Fig. 5 uses a covariance estimated from real data.

Fig. 2 shows an experiment using the covariance $C(\Delta) = \exp(-(15\Delta)^2)$. The low-resolution points are interpolated exactly.

Fig. 3 uses the oscillatory kernel $C(\Delta) = \exp(-\Delta^2)\cos(18\Delta)$ to interpolate randomly sampled data points. This figure illustrates an important difference between this form of superresolution and spline interpolation: while spline interpolation is generic (data agnostic) and minimizes a form of curvature, the superresolution is based on the covariance and can potentially emphasize curvature. In this case the chosen kernel forces wild oscillation.

Fig. 4 shows hole filling or "inpainting", in which a contiguous range of data are missing and filled in.

Fig. 5 shows inpainting of an approximately periodic signal obtained from motion capture of a joint angle (x-rotation of left tibia) from a walking motion. The data for this example was obtained from Subject #2 in [1].

# 6 Conclusion

Radial basis functions, principal component analysis, and Gaussian processes are widely used in computer graphics and computer vision. Particular applications include reconstruction of scanned data [4], motion capture from single-cameras [13], character animation [14], and many others. As each technique is founded on pairwise relationships expressed through a two-argument kernel or covariance

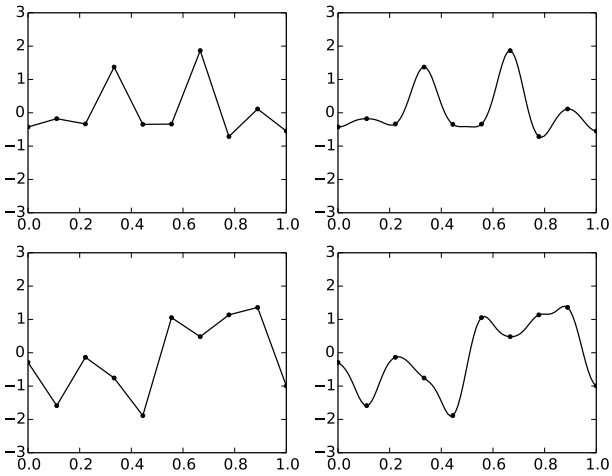

Figure 2: Left column: example signals synthesized with covariance $\exp(-(15\Delta)^2)$. Right column: corresponding superresolution signals. The indicated points are interpolated exactly.

function, it is not surprising that relationships between these techniques can be discovered. We illustrate the value of exploring these relationships through the derivation of a data-driven superresolution procedure.

# References


[1] Cmu graphics lab motion capture database. http://mocap.cs.cmu.edu, 2015.

[2] Ken Anjyo and J.P. Lewis. RBF interpolation and Gaussian process regression through an RKHS formulation. *JMI*, 3, 2011A-6:63–71, 2011.

[3] Ken Anjyo and J.P. Lewis. Scattered data interpolation for computer graphics. SIGGRAPH Course, http://portal.acm.org, 2014.

[4] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans. Reconstruction and representation of 3d objects with radial basis functions. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 67–76, New York, NY, USA, 2001. ACM.

[5] G. Casciola, L.B. Montefusco, and S. Morigi. Edge-driven image interpolation using adaptive anisotropic radial basis functions. *Journal of Mathematical Imaging and Vision*, 36(2):125–139, 2010.

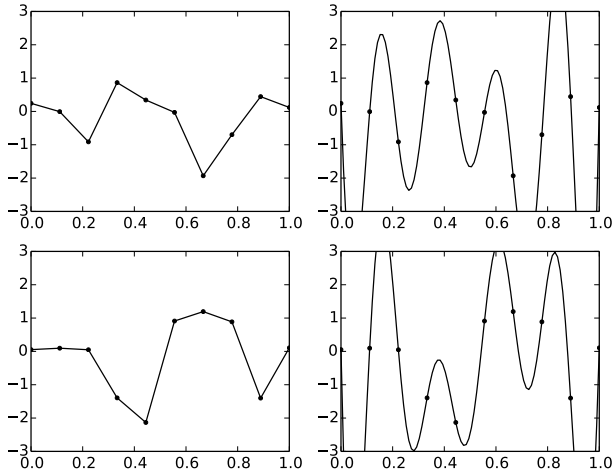


Figure 3: Left column: signals synthesized with kernel $\cos(18\Delta)\exp(-\Delta^2)$. Right column: corresponding superresolution signals. The indicated points are interpolated exactly. This kernel illustrates that the superresolution procedure is different from spline interpolation. In this case it is adding detail in the form of oscillation.

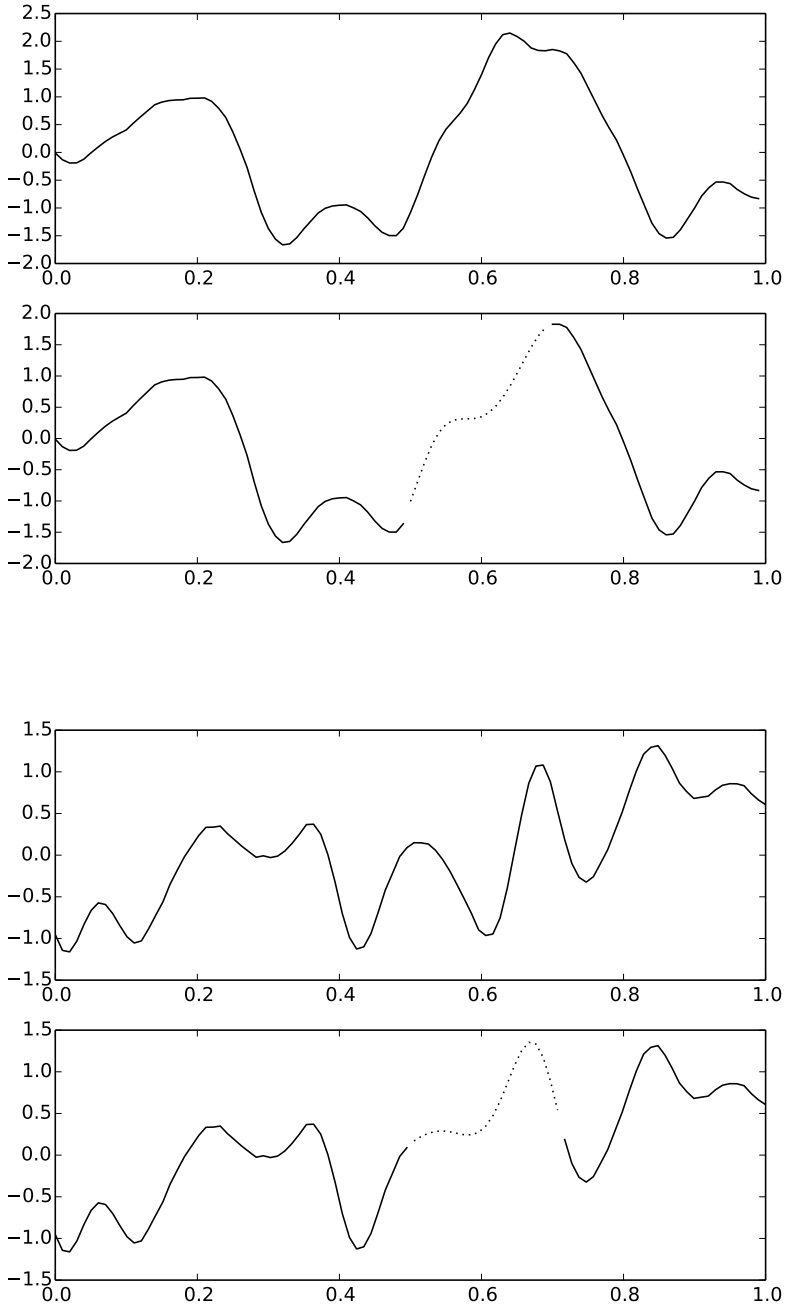


Figure 4: Inpainting of signals synthesized from the covariance $\exp(-(15\Delta)^2)$. The top plot in each subfigure is the original signal; the bottom plot is signal with an omitted section estimated.

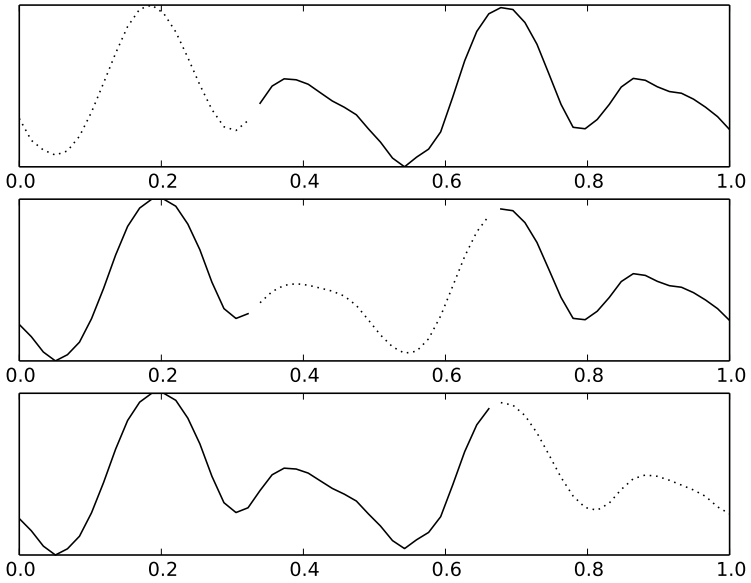


Figure 5: Inpainting of a joint angle signal from motion capture of a walking motion [1]. Each subplot shows one third of the signal omitted and estimated. The covariance is estimated from the original signal (Fig. 6).

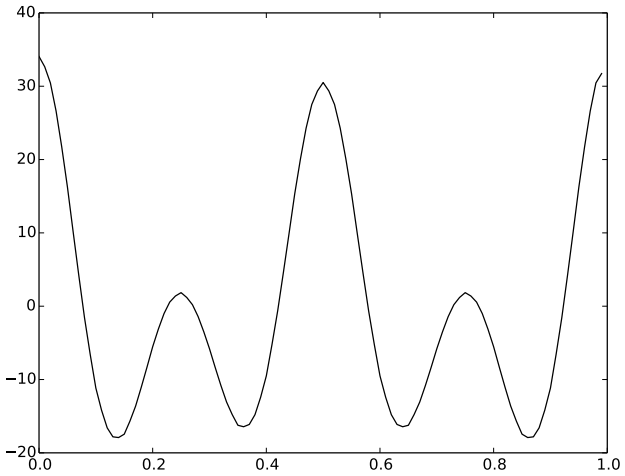


Figure 6: Covariance function estimated from the data used in Fig. 5.

[6] T. F. Cootes, G. J. Edwards, and C. J. Taylor. *Active Appearance Models*, volume 1407 of *Lecture Notes in Computer Science*. Springer, 1998.

[7] Huong Quynh Dinh, Greg Turk, and Greg Slabaugh. Reconstructing surfaces using anisotropic basis functions. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 606–613. IEEE, 2001.

[8] J.E. Gentle. *Random Number Generation and Monte Carlo Methods*. Statistics and Computing. Springer, 2003.

[9] P. J. Green and B. W. Silverman. *Nonparametric Regression and Generalized Linear Models: A Roughness Penalty Approach*. Chapman & Hall, London, 1994.

[10] M. Pourahmadi. *High-Dimensional Covariance Estimation: With High-Dimensional Data*. Wiley Series in Probability and Statistics. Wiley, 2013.

[11] K.R. Rao and P. Yip. *Discrete Cosine Transform: Algorithms, Advantages, Applications*. Academic Press, 1990.

[12] C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. Adaptive Computation And Machine Learning. Mit Press, 2006.

[13] Mathieu Salzmann and Raquel Urtasun. Implicitly constrained Gaussian process regression for monocular non-rigid pose estimation. In J.D. Lafferty, C.K.I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 2065–2073. Curran Associates, Inc., 2010.

[14] Yeongho Seol, Jaewoo Seo, Paul Hyunjin Kim, J. P. Lewis, and Junyong Noh. Weighted pose space editing for facial animation. *The Visual Computer*, 28(3):319–327, 2012.

[15] Tim Valentine. Face-space models of face recognition. In M.J. Wenger and J.T. Townsend, editors, *Computational, Geometric, and Process Perspectives on Facial Cognition: Contexts and Challenges*, Scientific Psychology Series. Taylor & Francis, 2012.