Super-resolution using Gaussian Process Regression

Final Year Project Interim Report

He He

Department of Electronic and Information Engineering
The Hong Kong Polytechnic Unviersity

December 30, 2010

Outline

1. Introduction
2. Gaussian Process Regression
   - Multivariate Normal Distribution
   - Gaussian Process
   - Regression
   - Training
3. GPR for Super-resolution
   - Framework
   - Covariance Function

Outline

1. Introduction
2. Gaussian Process Regression
   - Multivariate Normal Distribution
   - Gaussian Process
   - Regression
   - Training
3. GPR for Super-resolution
   - Framework
   - Covariance Function

The goal of **super-resolution (SR)** is to estimate a high-resolution (HR) image from one or a set of low-resolution (LR) images. It is widely applied in face recognition, medical imaging, HDTV etc.

Figure: Face recognition in video.

The goal of **super-resolution (SR)** is to estimate a high-resolution (HR) image from one or a set of low-resolution (LR) images. It is widely applied in face recognition, medical imaging, HDTV etc.

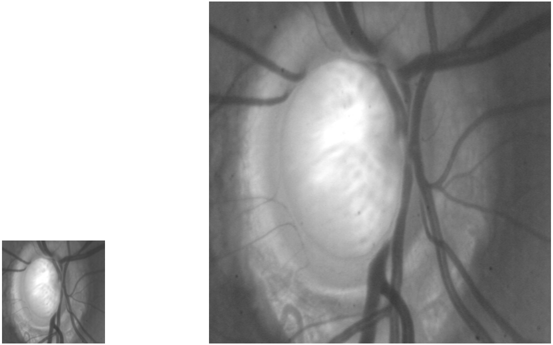


Figure: Super-resolution in medical imaging.

Super-resolution Methods

Interpolation-based methods

Fast but the HR image is usually blurred. E.g., bicubic interpolation, NEDI.

Learning-based methods

Hallucinate textures from the HR/LR image pair database.

Reconstruction-based methods

Formalize an optimization problem constrained by the LR image with various priors.

Outline

Multivariate Normal Distribution

Definition

A random vector $\mathbf{X} = (X_1, X_2, \ldots, X_p)$ is said to be multivariate normally (MVN) distributed if every linear combination of its components $\mathbf{Y} = \mathbf{a}^T\mathbf{X}$ has a univariate normal distribution. Real-world random variables can often be approximated as following a multivariate normal distribution.

The *probability density function* of $\mathbf{X}$ is

$$f(\mathbf{x}) = \frac{1}{(2\pi)^(p/2)|\mathbf{\Sigma}|^{1/2}} \exp\left\{\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\} \qquad (1)$$

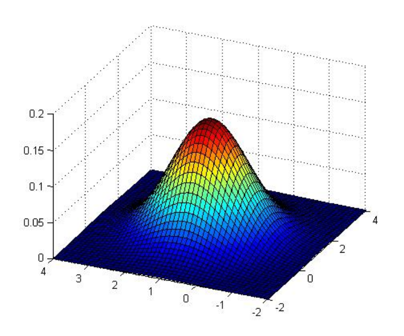
where $\boldsymbol{\mu}$ is the mean of $\mathbf{X}$ and $\mathbf{\Sigma}$ is the covariance matrix.

Multivariate Normal Distribution

Example

Bivariate normal distribution

$$\boldsymbol{\mu} = [1\ 1]', \ \boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$



Multivariate Normal Distribution

Property 1

The joint distribution of two MVN random variables is also an MVN distribution.

Given $\mathbf{X}_1 \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$, $\mathbf{X}_2 \sim \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$ and $\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \end{bmatrix}$, we have $\mathbf{X} \sim N_p(\mu, \boldsymbol{\Sigma})$ with $\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix}$, $\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{11} \end{bmatrix}$.

Multivariate Normal Distribution

Property 2
The conditional distribution of the components of MVN are (multivariate) normal.

The distribution of $\mathbf{X}_1$, given that $\mathbf{X}_2 = \mathbf{x}_2$, is normal and has

$$\text{Mean} = \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\mathbf{x}_2 - \boldsymbol{\mu}_2) \tag{2}$$

$$\text{Covariance} = \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21} \tag{3}$$

Gaussian Process

Definition

Gaussian Process (GP) defines a distribution over the function f, where f is a mapping from the input space $\mathcal{X}$ to $\mathfrak{R}$, such that for any finite subset of $\mathcal{X}$, its marginal distribution $P(f(\mathbf{x}_1), f(\mathbf{x}_2), ...f(\mathbf{x}_n))$ is a multivariate normal distribution.

$$\mathbf{f}|\mathbf{X} \sim \mathcal{N}(m(\mathbf{x}), K(\mathbf{X},\mathbf{X})) \tag{4}$$

where

$$\begin{aligned} \mathbf{X} &= \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} && (5) \\ m(\mathbf{x}) &= E[f(\mathbf{x})] && (6) \\ k(\mathbf{x}_i, \mathbf{x}_j) &= E\left[(f(\mathbf{x}_i) - m(\mathbf{x}))(f(\mathbf{x}_i)^T - m(\mathbf{x}^T))\right] && (7) \end{aligned}$$

and $K(\mathbf{X},\mathbf{X})$ denotes the covariance matrix such that $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$.

Gaussian Process

Formally, we write the Gaussian Process as

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}_i, \mathbf{x}_j)) \tag{8}$$

Without loss of generality, the mean is usually taken to be zero.

- Parameterized by the *mean* function $m(\mathbf{x})$ and the *covariance* function $k(\mathbf{x}_i, \mathbf{x}_j)$
- Infer in the function space directly

Gaussian Process Regression

Model:

$$f(\mathbf{x}) \quad \sim \quad \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}_i, \mathbf{x}_j)) \tag{9}$$

Given the inputs $\mathbf{X}_*$, the output $\mathbf{f}_*$ is

$$\mathbf{f}_* \sim \mathcal{N}(\mathbf{0}, K(\mathbf{X}_*, \mathbf{X}_*)) \tag{10}$$

According to the Gaussian prior, the joint distribution of the training outputs $\mathbf{f}$, and the test outputs $\mathbf{f}_*$ is

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(\mathbf{X}, \mathbf{X}) & K(\mathbf{X}, \mathbf{X}_*) \\ K(\mathbf{X}_*, \mathbf{X}) & K(\mathbf{X}_*, \mathbf{X}_*) \end{bmatrix}\right). \tag{11}$$

Noisy Model

In reality, we do not have access to true function values but rather noisy observations. Assuming independent indentically distributed noise, we have the noisy model

$$
\begin{aligned}
y &= f(\mathbf{x}) + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma_n^2) && (12) \\
f(\mathbf{x}) &\sim \mathcal{GP}(m(\mathbf{x}), K(\mathbf{X}, \mathbf{X})) && (13) \\
\mathrm{Var}(y) &= \mathrm{Var}(f(\mathbf{x})) + \mathrm{Var}(\varepsilon) = K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I && (14)
\end{aligned}
$$

Thus, the joint distribution for prediction is

$$
\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I & K(\mathbf{X}, \mathbf{X}_*) \\ K(\mathbf{X}_*, \mathbf{X}) & K(\mathbf{X}_*, \mathbf{X}_*) \end{bmatrix}\right) \qquad (15)
$$

Prediction

Referring to the previous property of the conditional distribution, we can obtain

$$\mathbf{f}_* \sim \mathcal{N}(\bar{\mathbf{f}}, V(\mathbf{f}_*)) \quad (16)$$

$$\bar{\mathbf{f}}_* = K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1}\mathbf{y}, \quad (17)$$

$$V(\mathbf{f}_*) = K(X_*, X_*) - K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1} K(X, X_*). \quad (18)$$

$\mathbf{y}$ are the training outputs and $\mathbf{f}_*$ are the test outputs, which are predicted as the mean $\bar{\mathbf{f}}$.

Marginal Likelihood

GPR model:

$$\mathbf{y} = \mathbf{f} + \epsilon \quad (19)$$

$$\mathbf{f} \sim \mathcal{GP}(m(\mathbf{x}), \mathbf{K}) \quad (20)$$

$$\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma_n^2 \mathbf{I}) \quad (21)$$

y is an n-dimensional vector of observations. Without loss of generality, let $m(\mathbf{x}) = 0$. Thus $y|\mathbf{X}$ follows a normal distribution with

$$E(y|\mathbf{X}) = 0 \quad (22)$$

$$Var(y|\mathbf{X}) = K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I} \quad (23)$$

Marginal Likelihood

Let $\mathbf{K}_y = Var(y|\mathbf{X})$,

$$p(\mathbf{y}|\mathbf{X}) = \frac{1}{(2\pi)^{n/2}|\mathbf{K}_y|^{1/2}} \exp\left\{-\frac{1}{2}\mathbf{y}^T\mathbf{K}_y^{-1}\mathbf{y}\right\} \tag{24}$$

The *log marginal likelihood* is

$$\mathcal{L} = \log p(\mathbf{y}|\mathbf{X}) = -\frac{n}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{K}_y| - \frac{1}{2}\mathbf{f}^T\mathbf{K}_y^{-1}\mathbf{f} \tag{25}$$

Maximum *a posteriori*

Matrix derivative:

$$\frac{\partial}{\partial x}\mathbf{Y} = -\mathbf{Y}^{-1}\frac{\partial \mathbf{Y}}{\partial \theta_i}\mathbf{Y}^{-1} \tag{26}$$

$$\frac{\partial}{\partial x}\log|\mathbf{Y}| = \operatorname{tr}(\mathbf{Y}^{-1}\frac{\partial \mathbf{Y}}{\partial \theta_i}) \tag{27}$$

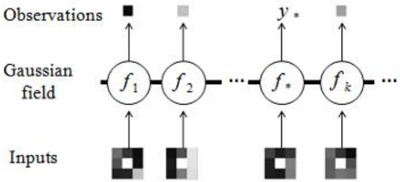
Gradient ascent:

$$\frac{\partial \mathcal{L}}{\partial \theta_i} = \frac{1}{2}\mathbf{y}^T K^{-1}\frac{\partial K}{\partial \theta_i}K^{-1}\mathbf{y} - \frac{1}{2}tr(K^{-1}\frac{\partial K}{\partial \theta_i}) \tag{28}$$

$\frac{\partial K}{\partial \theta_i}$ is a matrix of derivatives of each element.

Outline

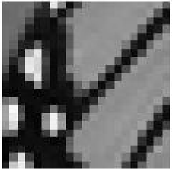
Graphical Representation



- Model: $y = f(\mathbf{x}) + \varepsilon$
- Squares: observed pixels
- Circles: unknown Gaussian field
- Inputs ($\mathbf{x}$): neighbors (predictors) of the target pixel
- Outputs (y): pixel at the center of each 3×3 patch
- Thick horizontal line: a set of fully connected nodes.

Stage 1: interpolation

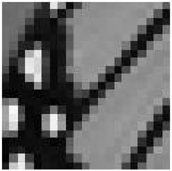
Input LR patch

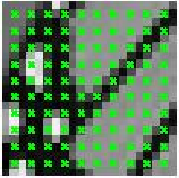


Workflow

Stage 1: interpolation

Sample training targets

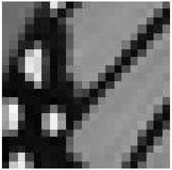


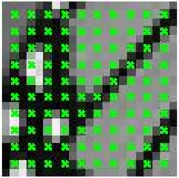


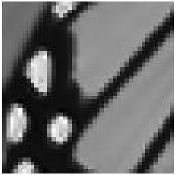
Workflow

Stage 1: interpolation

SR based on Bicubic Interpolation



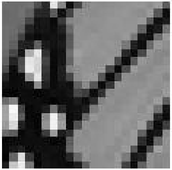


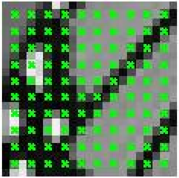


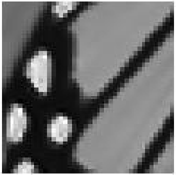
Stage 2: deblurring

Workflow

Stage 1: interpolation

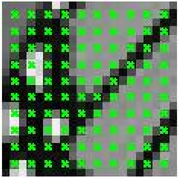






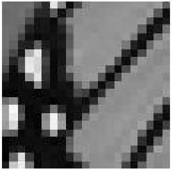
Stage 2: deblurring

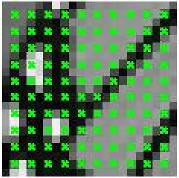
Sample training targets

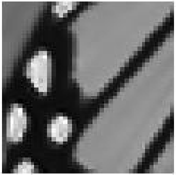


Workflow

Stage 1: interpolation

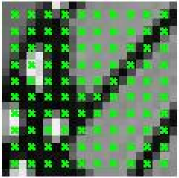


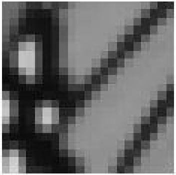




Stage 2: deblurring

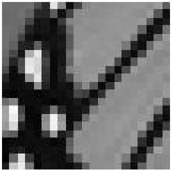
Obtain neighbors from the downsampled patch

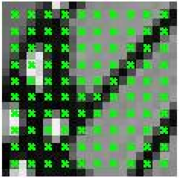


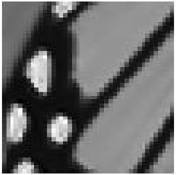


Workflow

Stage 1: interpolation

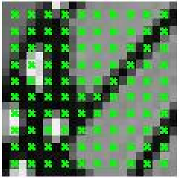


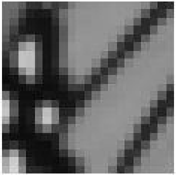




Stage 2: deblurring

SR based on the simulated blurring process





Covariance Equation

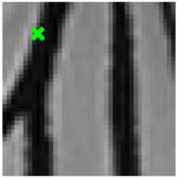
- defines the similarity between two points (vectors)
- indicate the underlying distribution of functions in GP
- Squared Exponential covariance function

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \exp\left(-\frac{1}{2}\frac{(\mathbf{x}_i - \mathbf{x}_j)'(\mathbf{x}_i - \mathbf{x}_j)}{\ell^2}\right) \quad (29)$$

σ_f^2 represents the signal variance and ℓ defines the *characteristic length scale*.

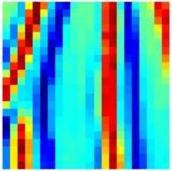
Given an image $\mathbf{I}$, the covariance between two pixels $\mathbf{I}_{i,j}$ and $\mathbf{I}_{m,n}$ is calculated as $k(\mathbf{I}_{(i,j),N}, \mathbf{I}_{(m,n),N})$, where N means to take the 8 nearest pixels around the pixel. Therefore, the similarity is based on the Euclidean distance between the pixels' neighbors.

Covariance Equation



(a) Test point

(b) Training patch



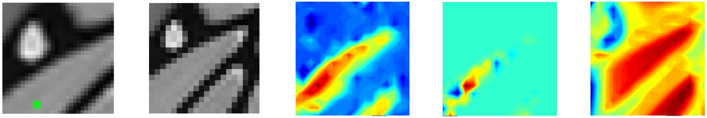
(c) Covariance matrix

- **Local similarity**: high responses (red regions) from the training patch are concentrated on edges
- **Global similarity**: high-responsive regions also include other similar edges within the patch
- **Conclusion**: pixels embedded in a similar structure to that of the target pixel in terms of the neighborhood tend to have higher weights during prediction

Hyperparameter Adaptation

Hyperparameters:

- σ_f^2: signal variance
- σ_n^2: noise variance
- ℓ: *characteristic length scale*



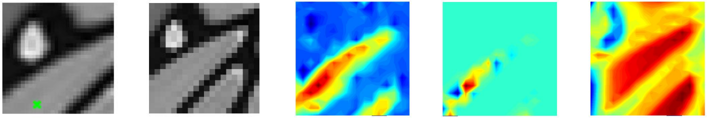
(a) Test (b) Training (c) $\ell = .50$, $\sigma_n = .01$ (d) $\ell = .05$, $\sigma_n = .001$ (e) $\ell = 1.65$, $\sigma_n = .14$

(c): MAP estimation
(d): Quickly varying field with low noise
(e): Slowly varyin field with high noise

Hyperparameter Adaptation

Hyperparameters:

- σ_f^2: signal variance
- σ_n^2: noise variance
- ℓ: *characteristic length scale*



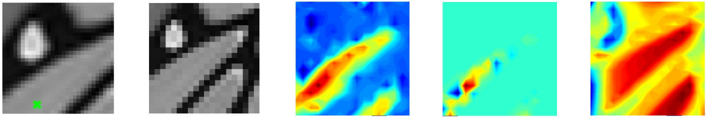
(a) Test (b) Training (c) $\ell = .50$, $\sigma_n = .01$ (d) $\ell = .05$, $\sigma_n = .001$ (e) $\ell = 1.65$, $\sigma_n = .14$

(c): MAP estimation
(d): Quickly varying field with low noise (high-frequncy artifacts)
(e): Slowly varyin field with high noise

Hyperparameter Adaptation

Hyperparameters:

- σ_f^2: signal variance
- σ_n^2: noise variance
- ℓ: *characteristic length scale*



(a) Test (b) Training (c) $\ell = .50$, $\sigma_n = .01$ (d) $\ell = .05$, $\sigma_n = .001$ (e) $\ell = 1.65$, $\sigma_n = .14$

(c): MAP estimation
(d): Quickly varying field with low noise (high-frequncy artifacts)
(e): Slowly varyin field with high noise (too smooth)

Hyperparameter Adaptation

Log marginal likelihood:

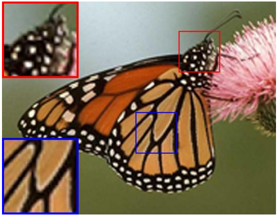
$$\log p(\mathbf{y}|X,\boldsymbol{\theta}) = -\frac{1}{2}\mathbf{y}^T K_y^{-1}\mathbf{y} - \frac{1}{2} log|K_y| - \frac{n}{2}\log 2\pi \tag{30}$$

Maximize *a posteriori* (gradient descent):

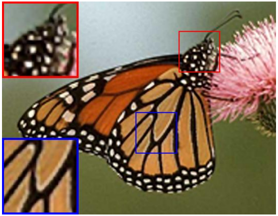
$$\frac{\partial \mathcal{L}}{\partial \theta_i} = \frac{1}{2}\mathbf{y}^T K^{-1}\frac{\partial K}{\partial \theta_i}K^{-1}\mathbf{y} - \frac{1}{2}tr(K^{-1}\frac{\partial K}{\partial \theta_i}) \tag{31}$$

$\boldsymbol{\theta}$ denotes the parameter set.

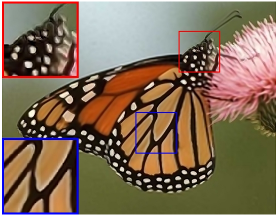
Results



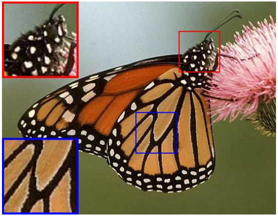
(a) Bicubic (MSSIM=0.84)



(b) GPP (MSSIM=0.84)

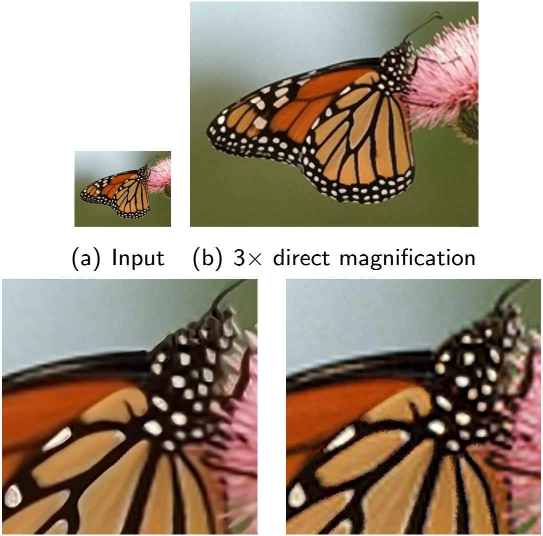


(c) Our result (MSSIM=0.86)



(d) Ground truth

Results



(a) Input (b) 3× direct magnification

(c) 10× our result (d) 10× detail synthesis

Results

(a) GPP

(b) Our result

Results

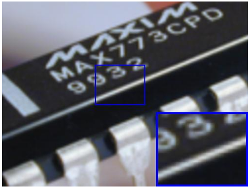
(a) Bicubic

(b) Edge statistics

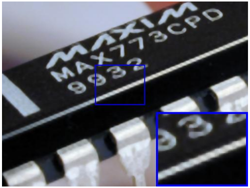
(c) Patch redundancy

(d) Ours

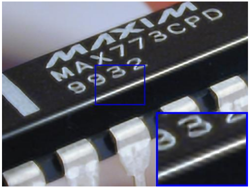
Results



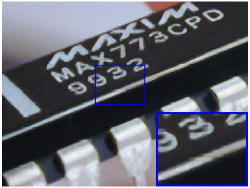
(a) Bicubic



(b) Edge statistics

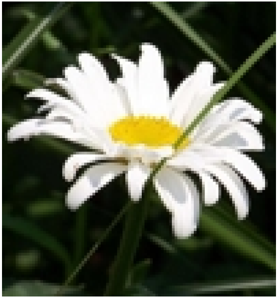


(c) Patch redundancy



(d) Ours

Results



(a) Bicubic

(b) Edge statistics

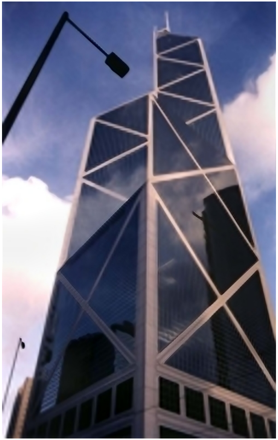
Results

(a) Bicubic

(b) Edge statistics

Results

(a) Bicubic



(b) Edge statistics