

CT5202: Photorealistic Rendering

Local Regression for Rendering Techniques

BOCHANG MOON

Local Regression

- A technique that models a relation between a predictor and response variable
- Statistical Model
 - $Y_i = f(x_i) + \epsilon_i$
 - $f(x_i)$: *unknown function*
 - Y_i : *observed value*
 - ϵ_i : *error*
 - Independent and identically distributed (IID)
 - $\text{var}(\epsilon_i) = \sigma_i$
 - Input: n pairs of predictor and response variables
 - $(x_1, Y_1), \dots, (x_n, Y_n)$
 - Output: estimated function $\hat{f}(x)$

Taylor Series

- Taylor expansion for 1D real function $f(x)$ for a range $[x_c - a < x_i < x_c + a]$
- $$f(x_i) \approx f(x_c) + \sum_{k=1}^p \frac{f^{(k)}(x_c)}{k!} (x_i - x_c)^k$$
- Taylor expansion for multi-dimensional functions $f(\mathbf{x})$
 - e.g., when $p = 1$ for 2D real function $f(x, y)$
 - $$f(x_i, y_i) \approx f(x_c, y_c) + (x_i - x_c) \frac{\partial f(x_c, y_c)}{\partial x} + (y_i - y_c) \frac{\partial f(x_c, y_c)}{\partial y}$$

Weight Function

- Weight function allocates higher weights to closer samples
 - $w\left(\frac{x_i - x_c}{h(x)}\right)$
 - Smoothing parameter (i.e., bandwidth) h controls the bias-variance tradeoff
 - e.g. tri-weight function
 - $w(u) = \begin{cases} (1 - |u|^3)^3, & |u| \leq 1 \\ 0, & |u| > 1 \end{cases}$
 - Weight function defines the local area where Taylor approximation is applied

Optimization Problem

- e.g., 1D Quadratic function
 - $f(x_i) \approx f(x_c) + (x_i - x_c)f'(x_c) + \frac{(x_i - x_c)^2}{2}f''(x_c)$
 - $f(x_i) \approx \beta_0 + (x_i - x_c)\beta_1 + \frac{(x_i - x_c)^2}{2}\beta_2$
- Computing the estimate $\hat{f}(x)$ is equivalent to find the $\beta = (\beta_0, \beta_1, \beta_2)$
 - $\hat{\beta} = \operatorname{argmin} \sum_{i=1}^n \left[Y_i - \left\{ \beta_0 + (x_i - x_c)\beta_1 + \frac{(x_i - x_c)^2}{2}\beta_2 \right\} \right]^2 w \left(\frac{x_i - x_c}{h(x)} \right)$

Normal Equation

- Closed form solution to the optimization problem

- $\hat{\beta} = (X^T W X)^{-1} X^T W Y$

- $$X = \begin{bmatrix} 1 & x_1 - x_c & \frac{(x_1 - x_c)^2}{2} \\ 1 & x_2 - x_c & \frac{(x_2 - x_c)^2}{2} \\ \dots & \dots & \dots \end{bmatrix}$$

- $$W = \begin{bmatrix} w \left(\frac{x_1 - x_c}{h(x)} \right) & 0 \\ 0 & w \left(\frac{x_1 - x_c}{h(x)} \right) \\ \dots & \dots \end{bmatrix}$$

- $$Y = [Y_1, \dots, Y_n]^T$$

Hat Matrix

- $\hat{f}(x) = X\hat{\beta} = X(X^T W X)^{-1} X^T W Y = H Y$
- $\hat{f}(x_c) = e_1' (X^T W X)^{-1} X^t W Y = \sum_{i=1}^n l_i(x_c) Y_i$
 - $e_1 = \left(1, x_c - x_c, \frac{(x_c - x_c)^2}{2}\right)^T = (1, 0, 0)^T$
 - Called a *linear smoother*
- H : hat matrix
 - Symmetric
 - Idempotent (i.e., $H H = H$)
- Residuals
 - $e = Y - \hat{f}(x) = Y - H Y = (I - H) Y$

Cross Validation

- Cross validation is a general technique that measures the error of estimated results
 - e.g., Leave-one-out cross-validation
 - $\hat{f}_{(-i)}(x_i) = \sum_{j \neq i} \frac{H_{ij} Y_j}{1 - H_{ii}}$
 - Reconstructed function without $(x_i, f(x_i))$ at x_i
 - Note that we don't need to re-run the normal equation
 - Predicted residual error sum of squares (PRESS)
 - $\sum_{i=1}^n (Y_i - \hat{f}_{(-i)}(x_i))^2$

Applications

- Image reconstruction
 - Input: predictors (e.g., image coordinates, optionally features) and response variables (e.g., noisy colors)
 - Output: reconstructed colors $\hat{f}(x)$



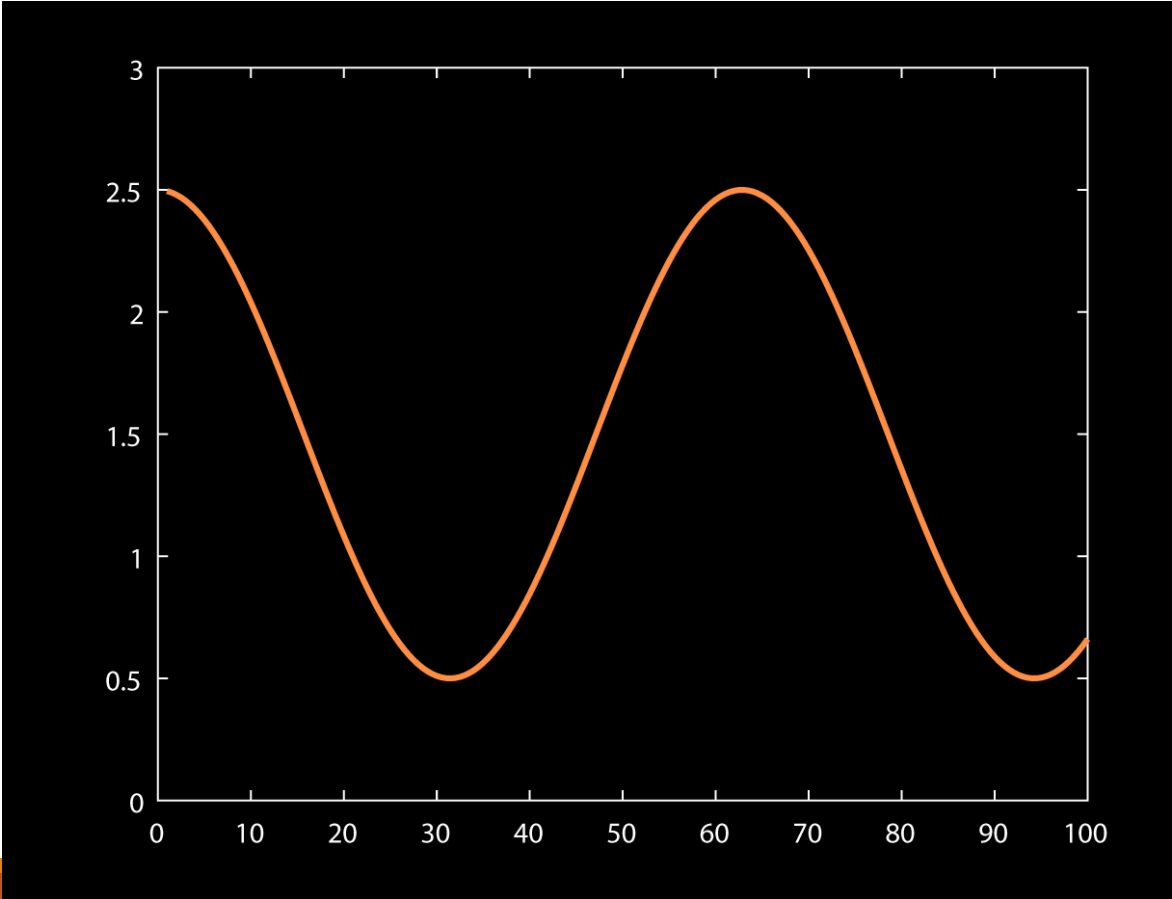
44 sec.

8 rays / pixel

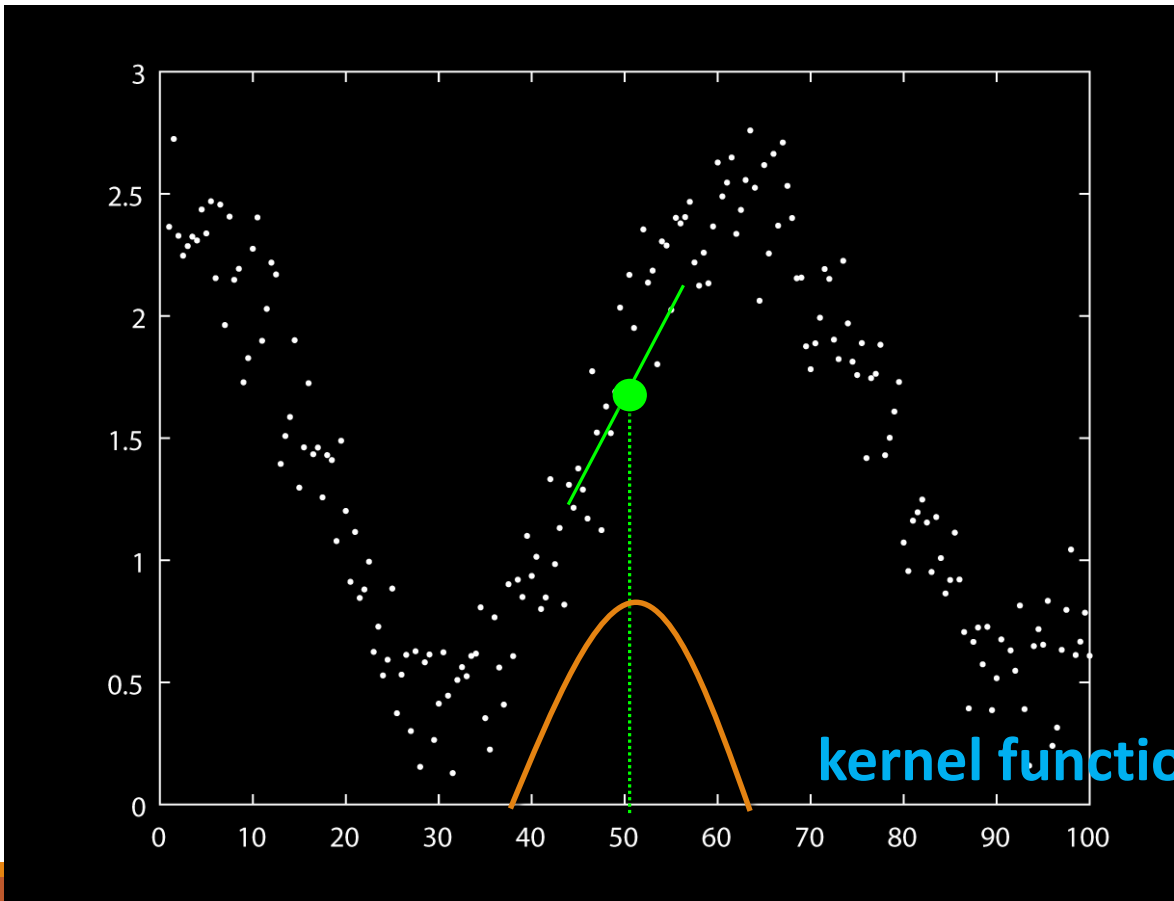


46 sec.

8 rays / pixel

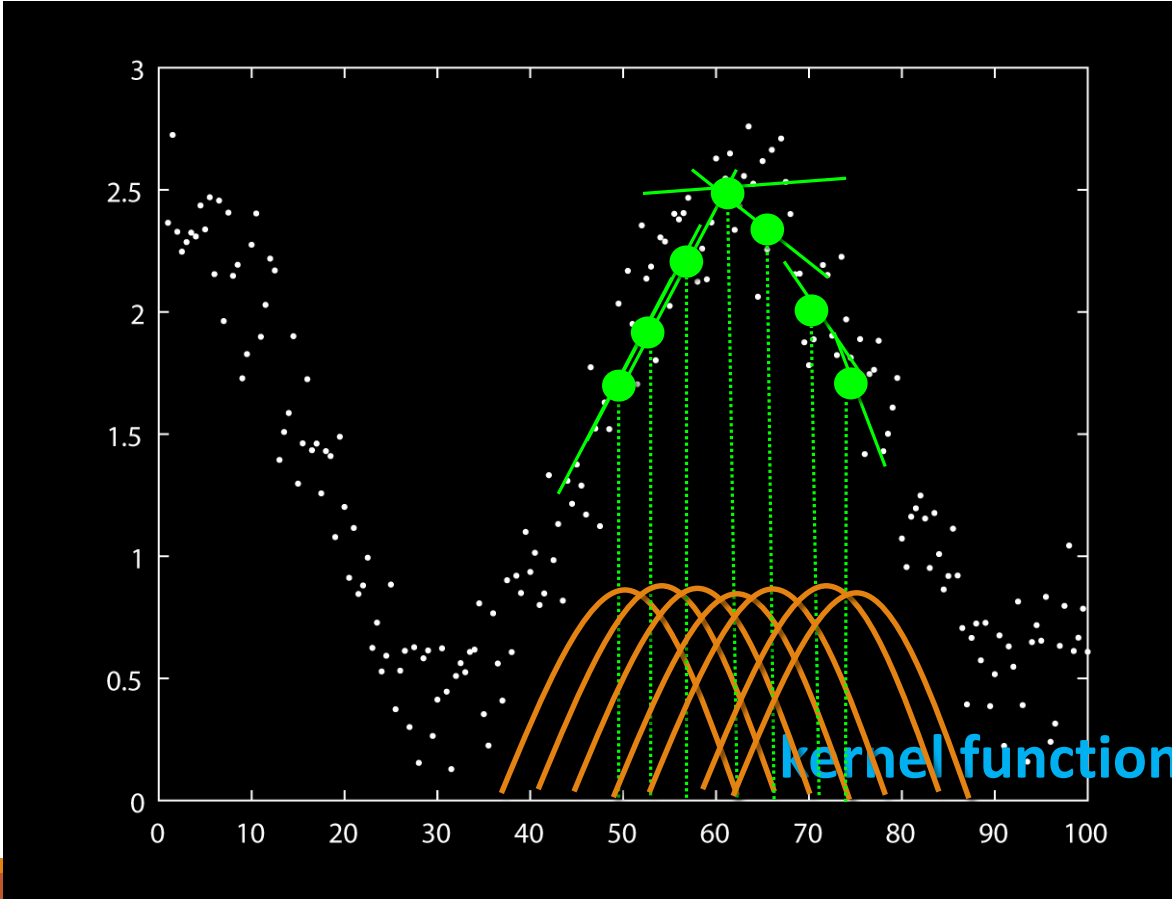


— Ground truth

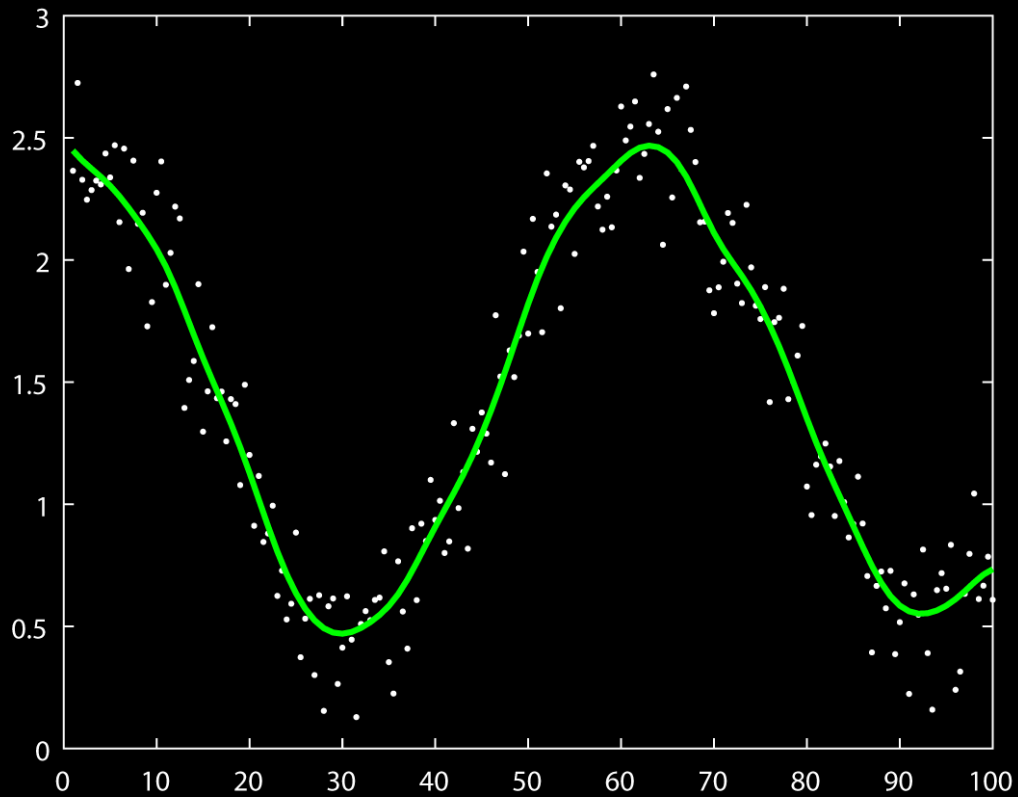


● Noisy input

kernel function

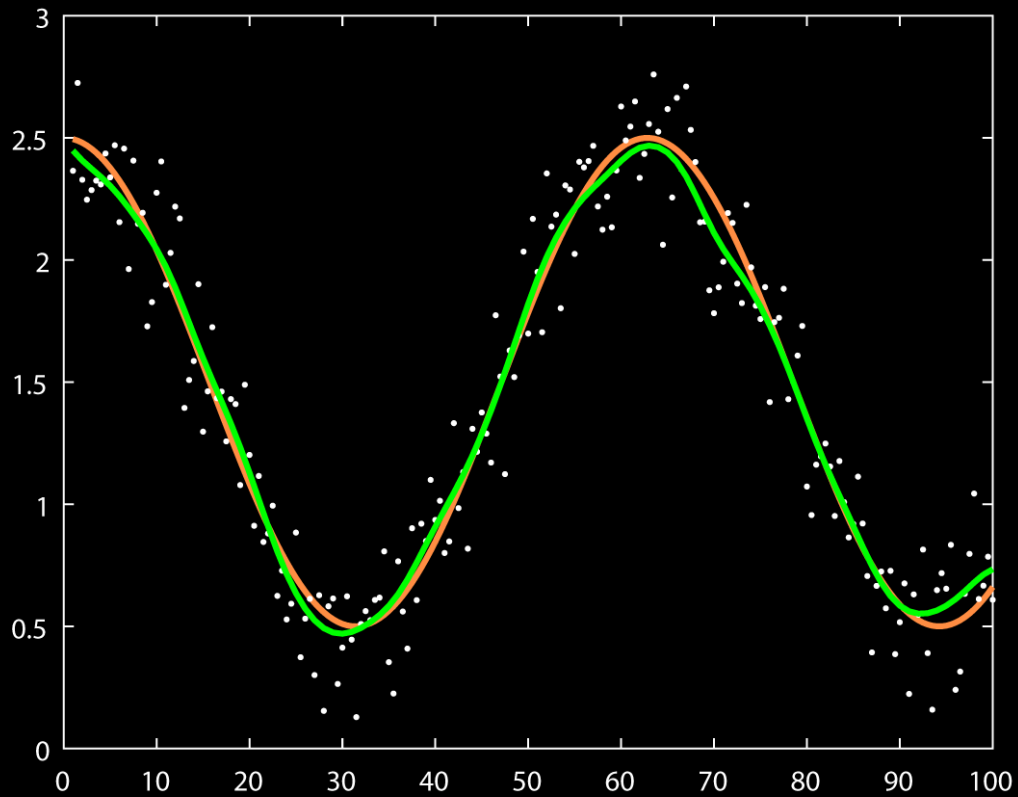


● Noisy input



● Noisy input

— Local regression



● Noisy input

— Local regression

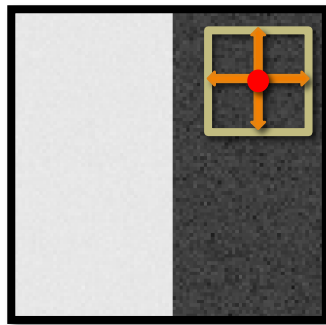
— Ground truth

Applications

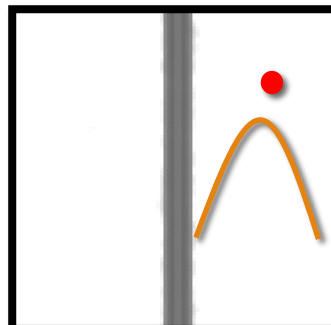
- Gradient (or high-order derivatives) estimation
 - e.g., estimated $f'(x_c)$ or $f''(x_c)$
- Hole filling
 - Predict $f(x_k)$ where we do not have any data at x_k

Discussion

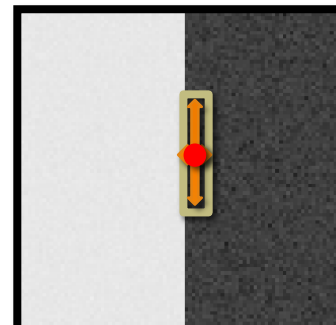
- Bandwidth



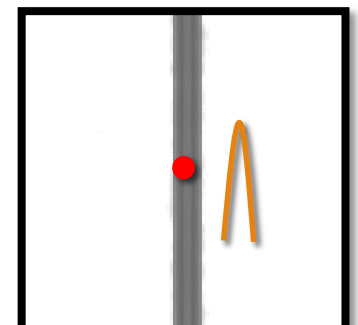
Input



Parameters



Input



Parameters

- Polynomial order

