Lecture 6 Image Super Resolution
(Conventional Methods)

Yuyao Zhang PhD
zhangyy8@shanghaitech.edu.cn
SIST Building-3 420

上海科技大学
ShanghaiTech University

Outline

- **Interpolation methods**
- **Reconstruction based methods**
- **Deep learning based methods**

Resource： https://github.com/wenbihan/reproducible-image-denoising-state-of-the-art

What is Super Resolution?

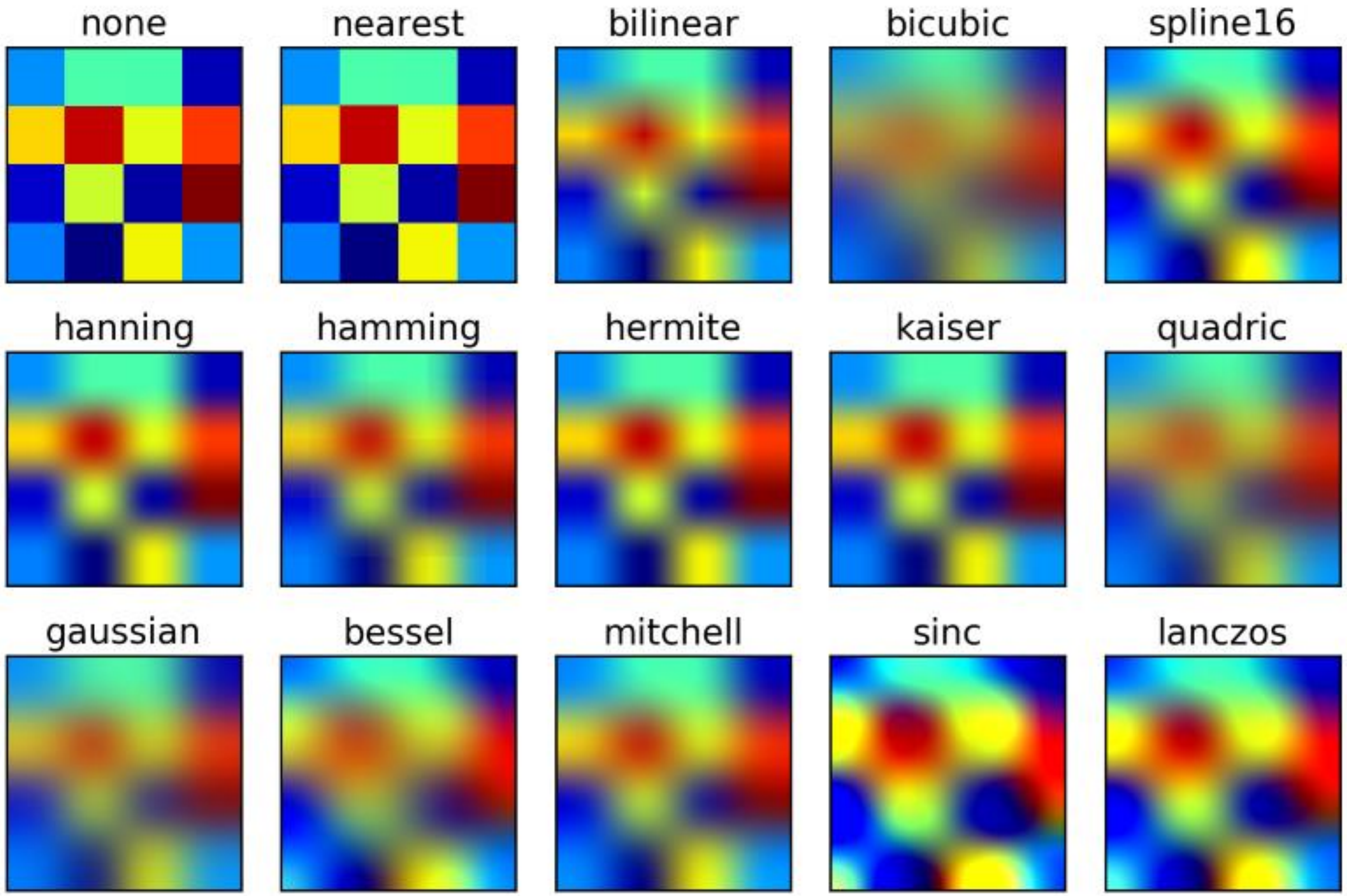
- **Super Resolution**
  - **Restore High-Resolution (HR) image (or video) from Low-Resolution (LR) image (or video).**
  - **According to number of input LR images, SR can be classified as single image super resolution (SISR) and multi-image super resolution (MISR).**

Super
Resolution

Interpolation methods

Interpolation based Image Super Resolution

- Well-known interpolation methods include：

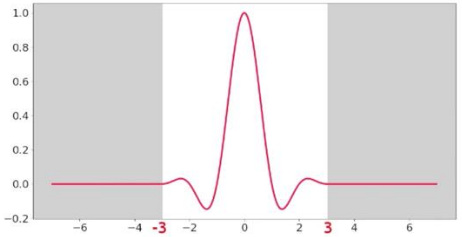


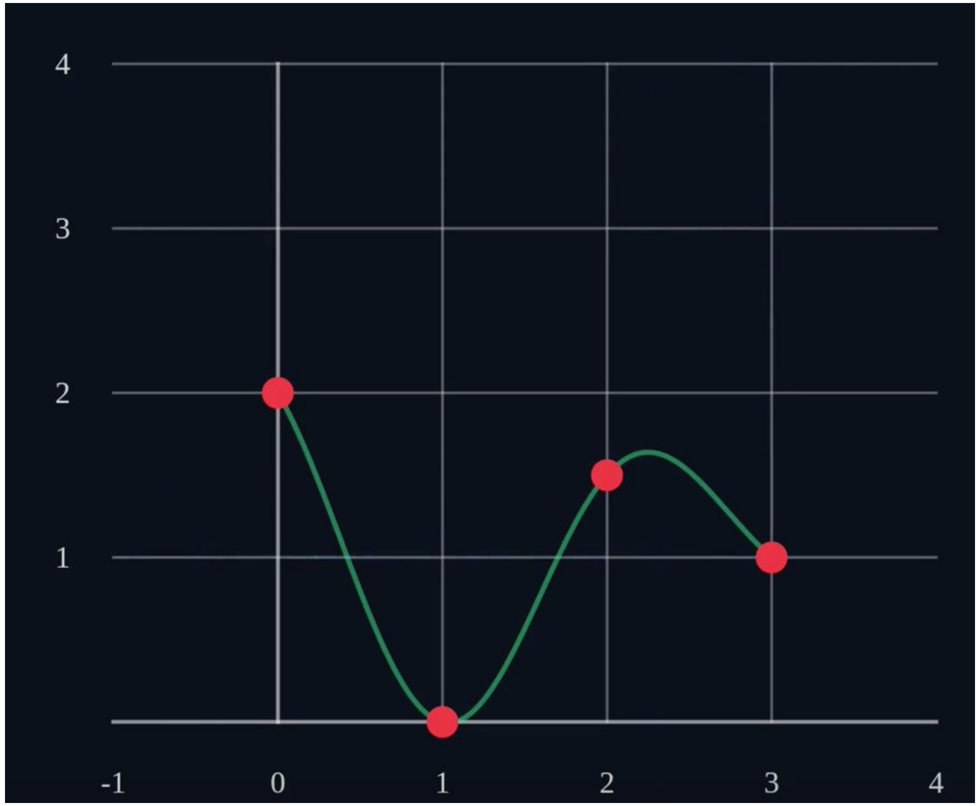
Lanczos Interpolation

$$L(x) = \begin{cases} 1 & if\ x = 0 \\ \dfrac{a\sin(\pi x)\sin(\pi x/a)}{\pi^2 x^2} & if\ -a \le x \le a \\ 0 & otherwise \end{cases}$$

$x = 0,1,2,3;\ S(\mathrm{x}) = (2,0,1.5,1)$

Where a is a positive integer. Typically between 2 and 3. Defined a function window as on the right figure.

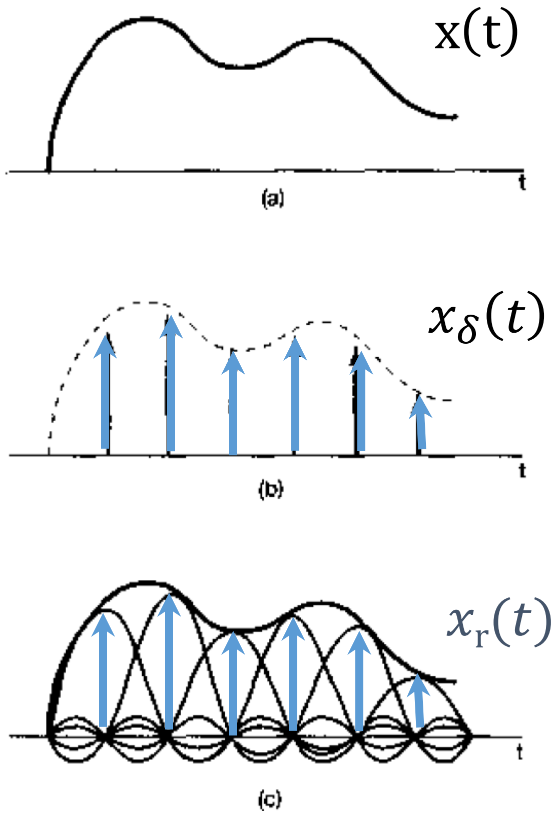




$$f(x) = \sum_{i=\lfloor x \rfloor - a + 1}^{\lfloor x \rfloor + a} S_i L(x - i)$$

Ideal Band-Limited Interpolation (Sinc Interpolation)

- What happened when HR image is down-sampled to LR?



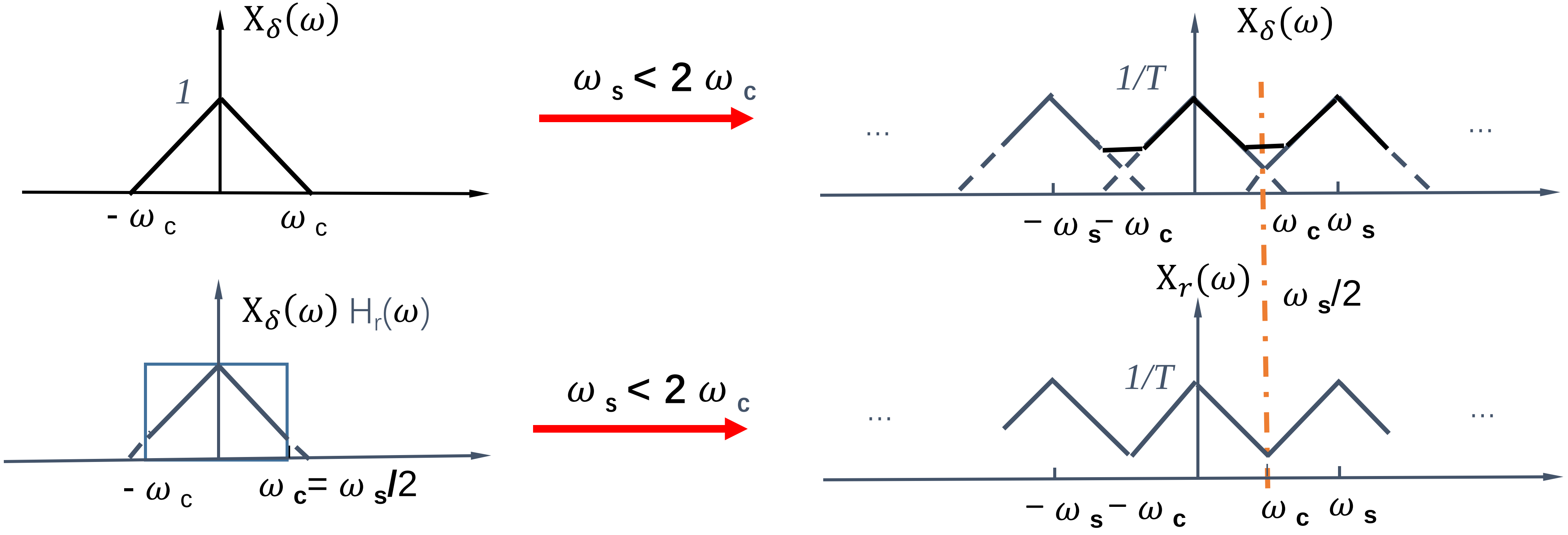
$$x_\delta(t) = \mathrm{x(t)} \cdot \sum_{-\infty}^{\infty} \delta(t - nT_s)$$

$$= \sum_{-\infty}^{\infty} x(nT_s)\delta(t - nT_s)$$

$$x_r(t) = ?$$

Ideal Band-Limited Interpolation (Sinc Interpolation)

- What happened when HR image is down-sampled to LR?



Ideal Band-Limited Interpolation (Sinc Interpolation)

- So to restore the HR image from down-sampled LR image, the filter below that will remove the aliasing artifact has a frequency response of

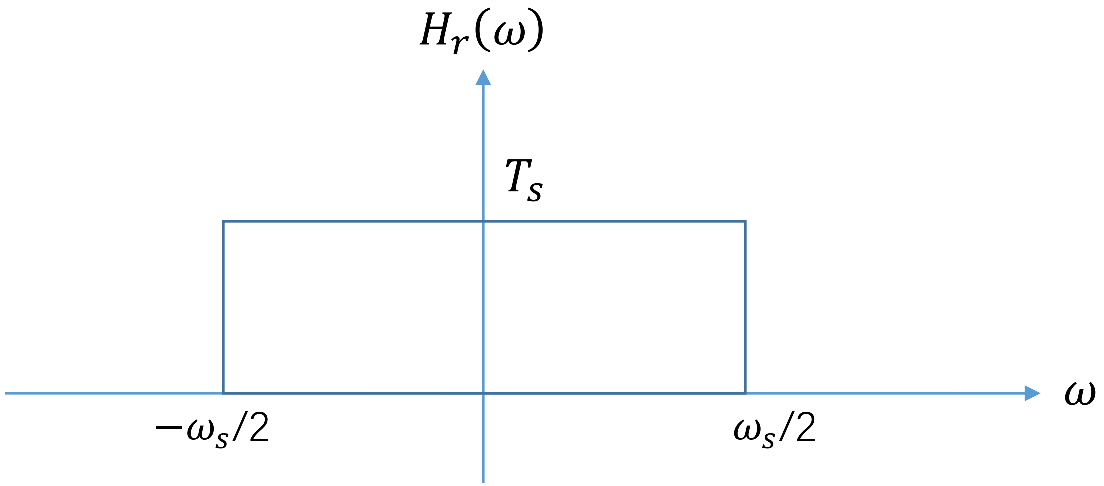
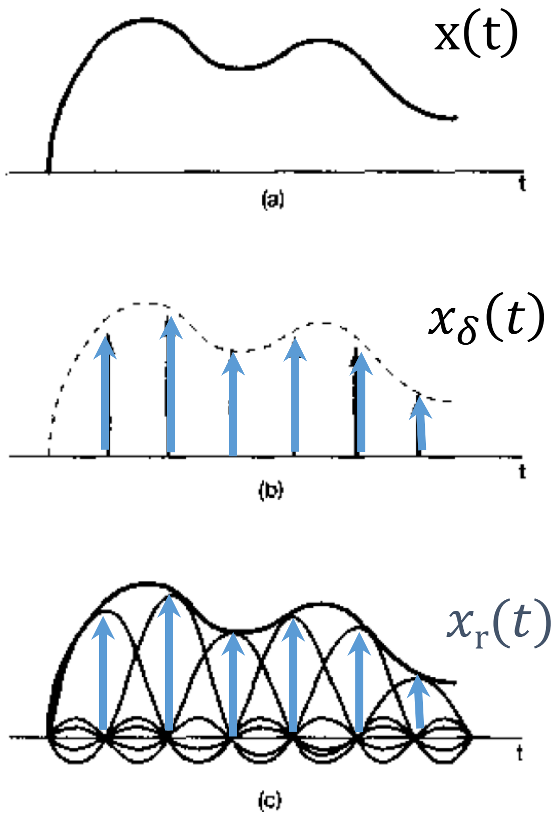


Figure: Frequency Response of Ideal Reconstruction Filter

Ideal Band-Limited Interpolation (Sinc Interpolation)

- Recover original spectrum:



$$X_r(\omega) = H_r(\omega) \cdot X_\delta(\omega)$$

$$\Rightarrow x_r(t) = h_r(t) * x_\delta(t)$$

$$x_\delta(t) = x(t) \cdot \sum_{-\infty}^{\infty} \delta(t - nT_s)$$

$$= \sum_{-\infty}^{\infty} x(nT_s)\delta(t - nT_s)$$

$$H_r(\omega) = T_s \mathrm{rect}\left(\frac{\omega}{2\frac{\omega_s}{2}}\right)$$

$$\Rightarrow h_r(t) = T_s \frac{\omega_s}{2\pi} sinc\left(\frac{\omega_s t}{2}\right)$$

F.T. table: $\mathrm{rect}\left(\frac{\omega}{2a}\right) \Leftrightarrow \frac{a}{\pi} sinc(at)$

$$sinc(x) = \frac{\sin(x)}{x}$$

Reconstruction based methods

Super-Resolution Through Neighbor Embedding

Hong Chang, Dit-Yan Yeung and Yimin Xiong

Presented By:
Ashish Parulekar, Ritendra Datta, Shiva Kasiviswanathan and Siddharth Pal

Hong Chang, Dit-Yan Yeung and Yimin Xiong, "Super-resolution through neighbor embedding," *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, Washington, DC, USA, 2004, pp. I-I, doi: 10.1109/CVPR.2004.1315043.

上海科技大学
ShanghaiTech University

Problem Formulation

img1.jpg

Training Xs^i

Training Ys^i

Testing Xt

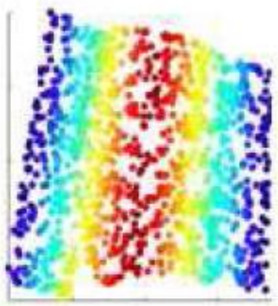
?

Testing Yt

Intuition

- Patches of the image lie on a manifold

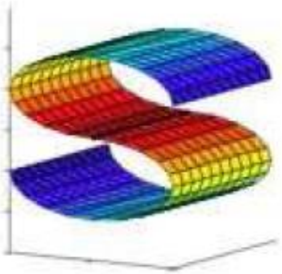
Training Xs^i



Low dimensional Manifold

img1.jpg

Training Ys^i



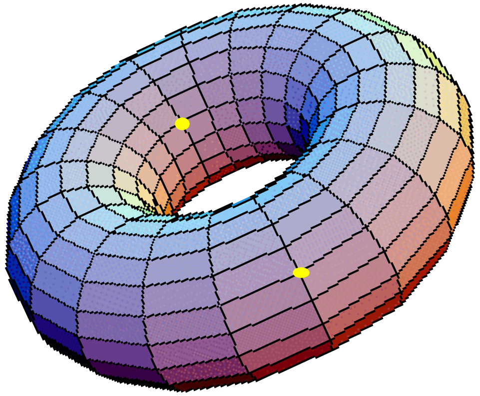
High dimensional Manifold

ShanghaiTech University 上海科技大学

Algorithm

1. Get feature vectors for each low resolution training patch.
2. For each test patch feature vector find K nearest neighboring feature vectors of training patches.
3. Find optimum weights to express each test patch vector as a weighted sum of its K nearest neighbor vectors.
4. Use these weights for reconstruction of that test patch in high resolution.

Manifold Definition Revisited



- A manifold is a topological space which is locally Euclidean.

- Represents a very useful and challenging unsupervised learning problem.

- In general, any object which is nearly "flat" on small scales is a manifold.

Manifold Learning

- Discover low dimensional structures (smooth manifold) for data in high dimension.

- Linear Approaches
  - Principal component analysis.
  - Multi dimensional scaling.

- Non Linear Approaches
  - Local Linear Embedding
  - ISOMAP
  - Laplacian Eigenmap.

Local Linear Embedding （LLE）

- Neighborhood preserving embeddings.
- Mapping to global coordinate system of low dimensionality.
- Recovering global nonlinear structure from locally linear fits.
- Each data point and it’s neighbors is expected to lie on or close to a locally linear patch.
- Each data point is constructed by it’s neighbors:

$$\hat{\vec{X}}_i = \sum_j W_{ij} \vec{X}_j$$

$$W_{ij} = 0 \text{ if } \vec{X}_j \text{ is not a neighbor of } \vec{X}_i$$

- Where W_{ij} summarize the contribution of j-th data point to the i-th data reconstruction and is what we will estimated by optimizing the error.
- Reconstructed from only its neighbors.

Local Linear Embedding （LLE）

- **We want to minimize the error function**

$$\varepsilon(W) = \sum_i \left| \vec{X}_i - \sum_j W_{ij} \vec{X}_j \right|^2$$

- **With the constraints :**

$$W_{ij} = 0 \ \text{ if } \vec{X}_j \text{ is not a neighbor of } \vec{X}_i$$

$$\sum_j W_{ij} = 1$$

- **Solution (using lagrange multipliers):**

$$W_j = \sum_k C_{jk}^{-1} (\vec{X}\vec{\eta}_k + \lambda)$$

$$\lambda = 1 - \sum_{jk} C_{jk}^{-1} (\vec{X}\vec{\eta}_k) \Big/ \sum_{jk} C_{jk}^{-1}$$

Local Linear Embedding （LLE）

- **Choose d-dimensional coordinates, Y, to minimize:** $$\phi(Y) = \sum_i \left| \vec{Y}_i - \sum_j W_{ij} \vec{Y}_j \right|^2$$
- **Under :** $$\sum_i \vec{Y}_i = \vec{0}, \quad \frac{1}{N} \sum_i \vec{Y}\vec{Y}^T = I$$

Quadratic form: $$\phi(Y) = \sum_{\mathrm{ij}} \mathrm{M}_{\mathrm{ij}} (\vec{Y}_i \vec{Y}_j)$$

where: $$M = (I - W)^T (I - W)$$

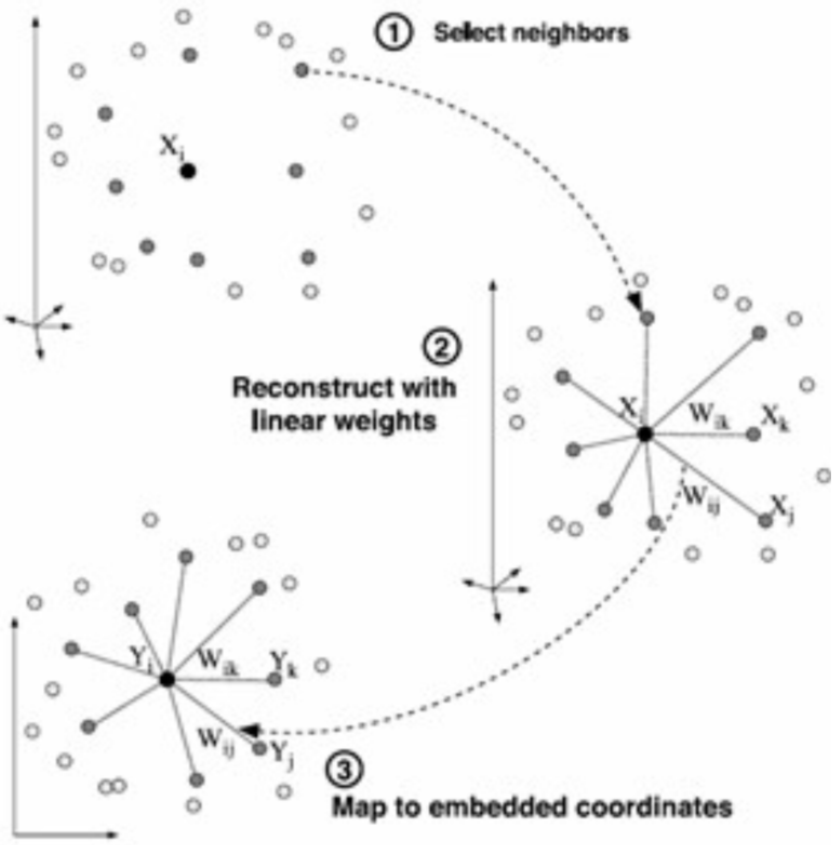
- **Solution : compute bottom d+1 eigenvectors of M. (discard the last one)**

- **Step 1**
  - Compute the neighbors of each data point, X_i
- **Step 2**
  - Compute the weight W_{ij} that best reconstruct each data point X_i from its neighbors, minimizing the cost in eq(1) by constrainted linear fits.

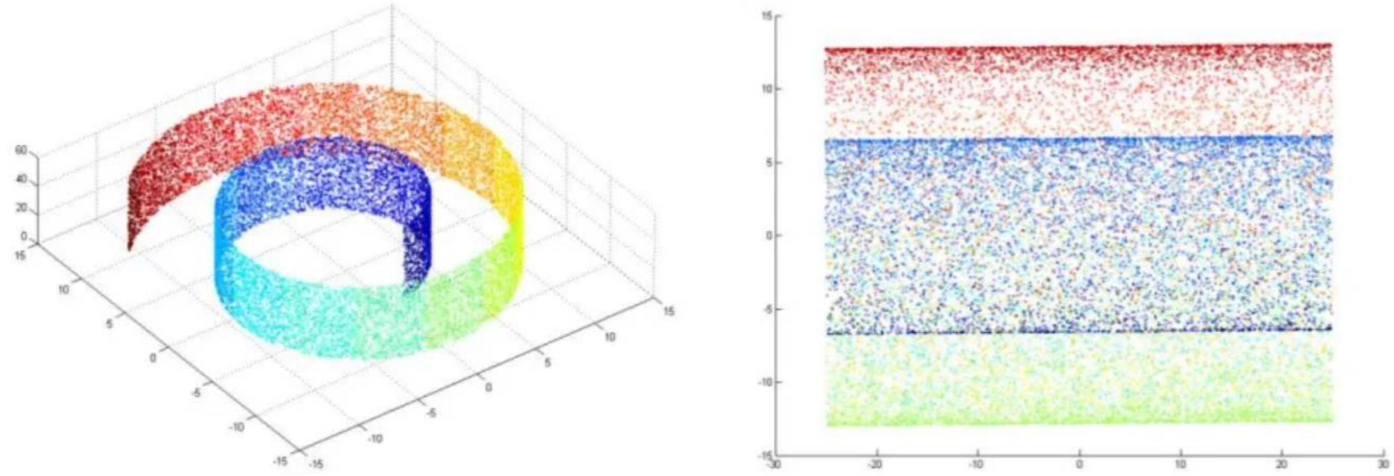
(1) $$\varepsilon(W) = \sum_i \left| \vec{X}_i - \sum_j W_{ij} \vec{X}_j \right|^2$$

- **Step 3**
  - Compute the vectors Y_i best reconstructed by the weights W_{ij}, minimizing the quadratic form in eq(2) by its bottom nonzero eigenvectors.

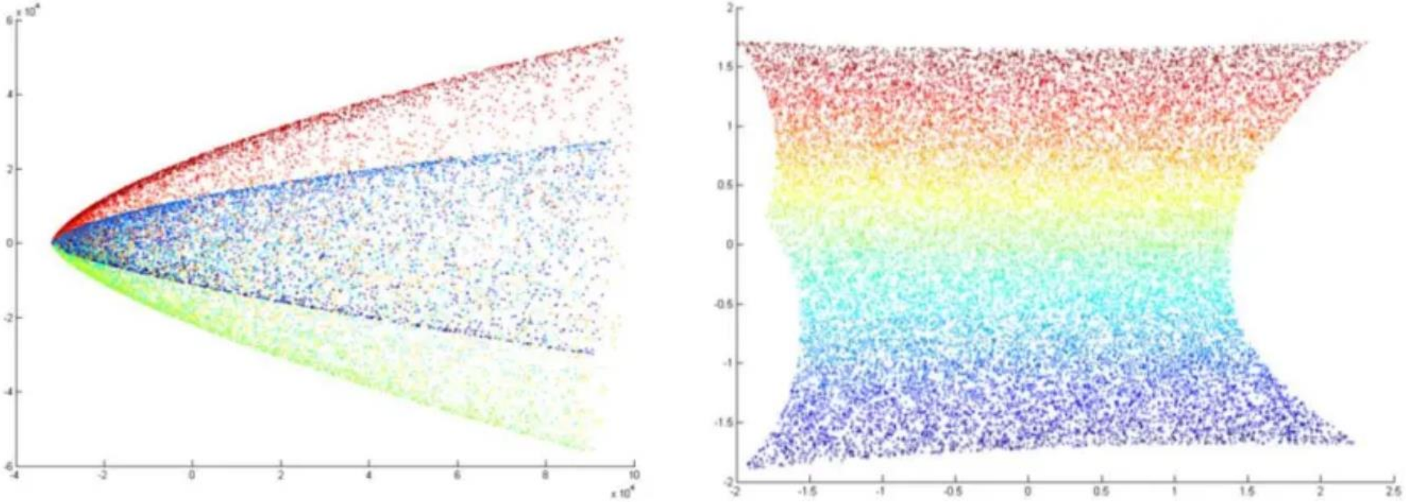
(2) $$\phi(Y) = \sum_i \left| \vec{Y}_i - \sum_j W_{ij} \vec{Y}_j \right|^2$$



Local Linear Embedding （LLE）



From left : Swiss Roll features in 3 dimensions, Swiss Roll example using PCA Source



Problem Formulation

Training Xs^i

img1.jpg

Training Ys^i

Testing Xt

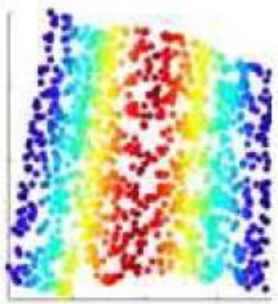
?

Testing Yt

Intuition

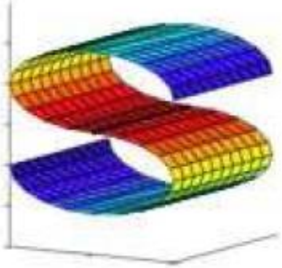
- Patches of the image lie on a manifold

Training Xs^i



Low dimensional Manifold

Training Ys^i



High dimensional Manifold

上海科技大学
ShanghaiTech University

Experimental Setup

- Target magnification : 3X
- Transform to YIQ space
- Feature Selection
  - First derivative of luminance
  - Second derivative of luminance
- Reconstruction of Y and transfer of I and Q from lower dimension.

Results

Experiments with images from the paper.

Intuition: Authors' test set may be biased !

Results

Training Xs^i

img1.jpg

Training Ys^i

Testing Xt

Testing Yt

上海科技大学 ShanghaiTech University

Results

Training Xs^i

Training Ys^i

Testing Xt

Testing Yt

上海科技大学 ShanghaiTech University

Results

Training Xs^i

Training Ys^i

Testing Xt

Testing Yt

Results

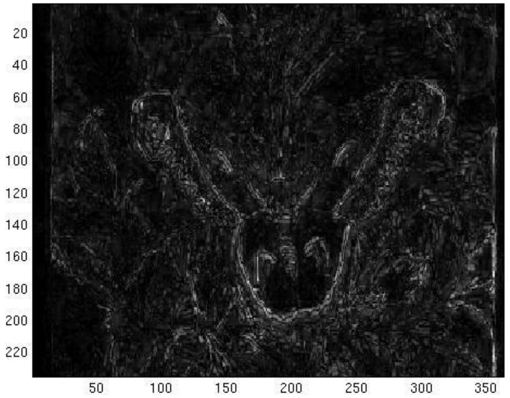
Training Xs^i

Training Ys^i

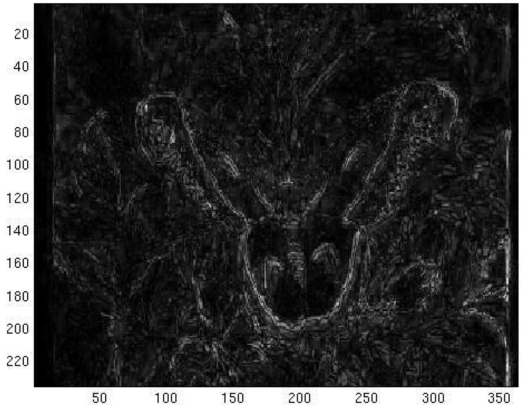
Testing Xt

Testing Yt

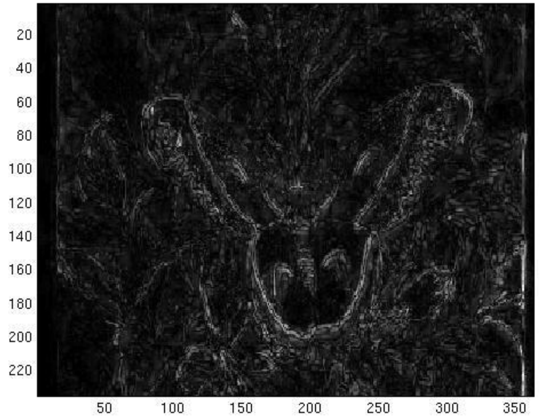
Results



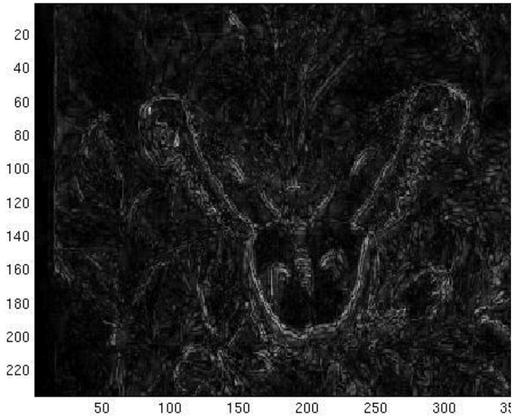
K=1



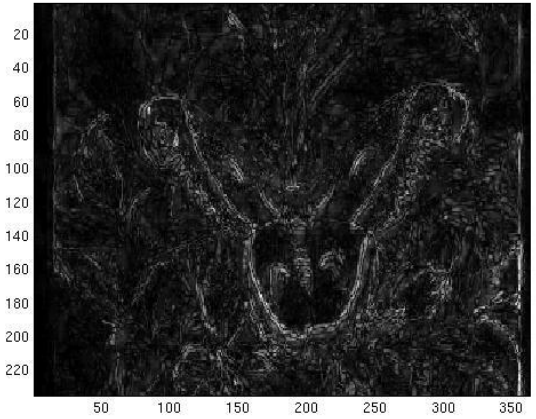
K=2



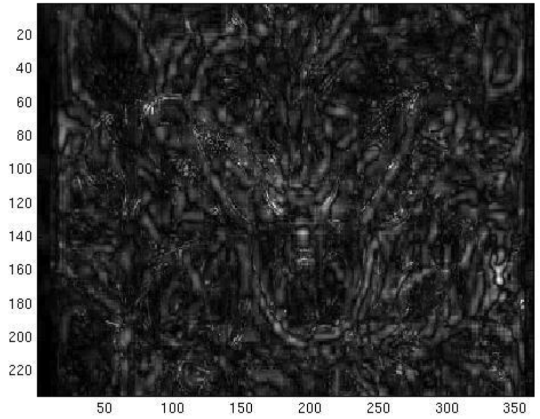
K=3



K=4



k=5



k=6

Results

As expected, good super-resolution images generated.

Now for the real test – our images !

Results – *their method our data*

Ground Truth Image

Our Test Image

1/3X

Obtained from scaling down

Results – *their method our data*

High-resolution training images : (A) Similar (B) Dissimilar images

Corresponding low-resolution training image

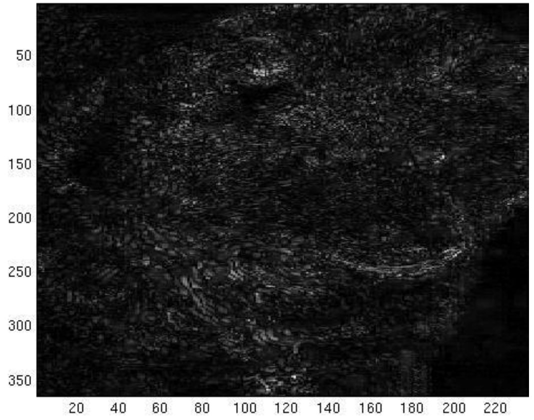
Results – *their method our data*

We test the results using RMS Error using the following formula:

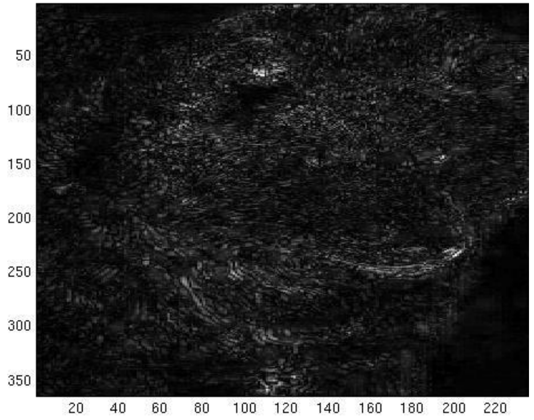
$$\text{Error} = \sqrt{\ } \ (1/n) \sum || P_{\text{ground-truth}} - P_{\text{generated}} ||$$

This essentially indicates the average Luminance deviations between corresponding pixels in the Ground-truth and Generated Images. Remember, the Chrominance components I and Q were copied into the generated image without change.

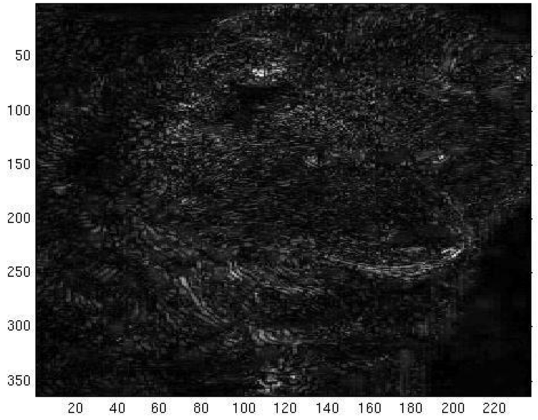
Results – *Similar training Image*



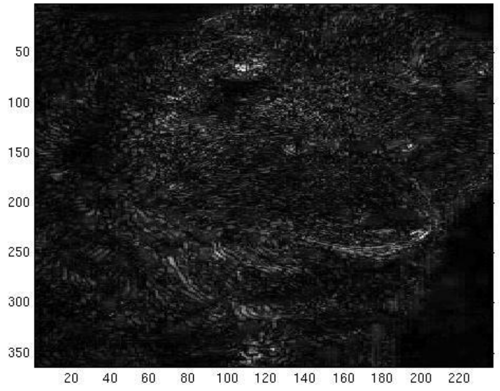
K=1



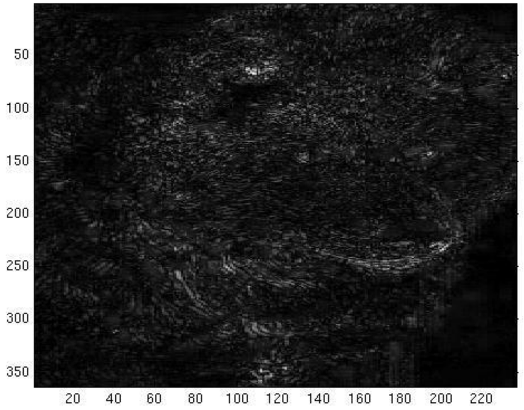
K=2



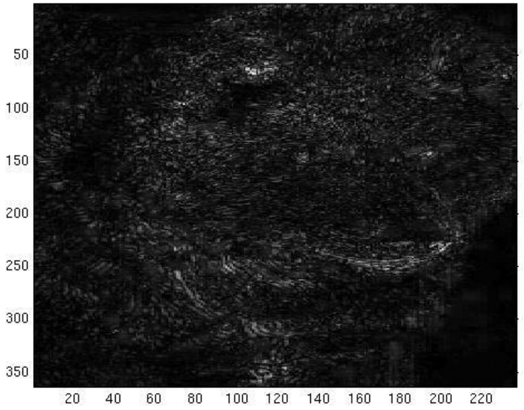
K=3



K=4



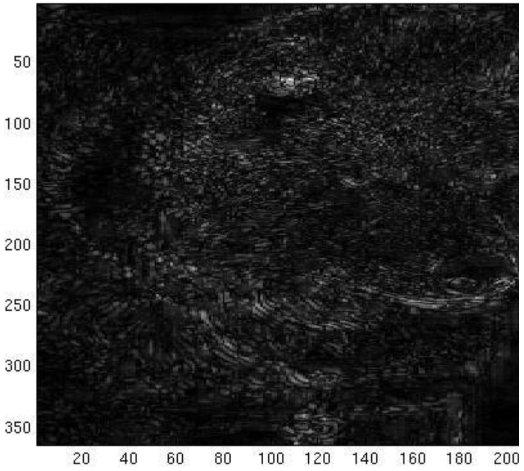
K=5



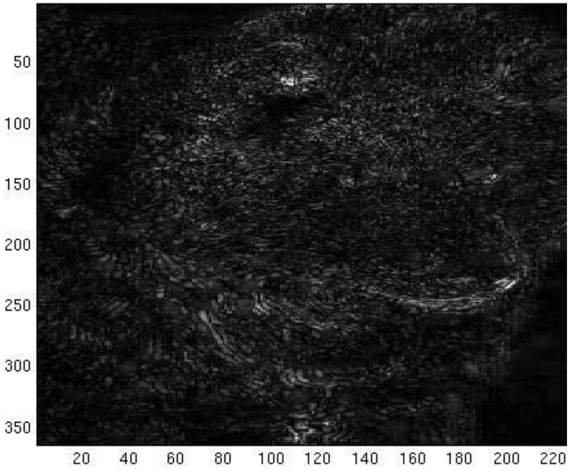
K=6

上海科技大学
ShanghaiTech University

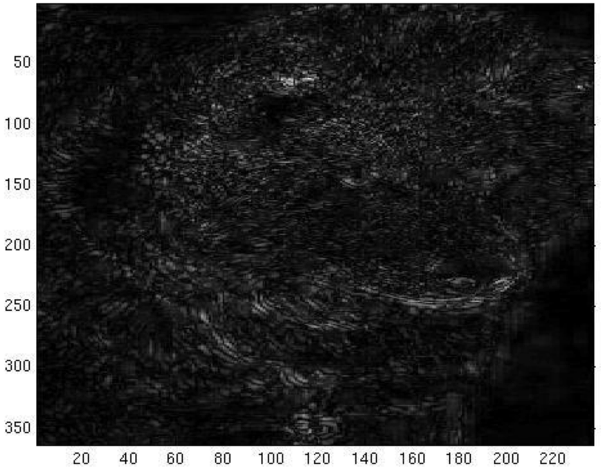
Results – *Different training Image*



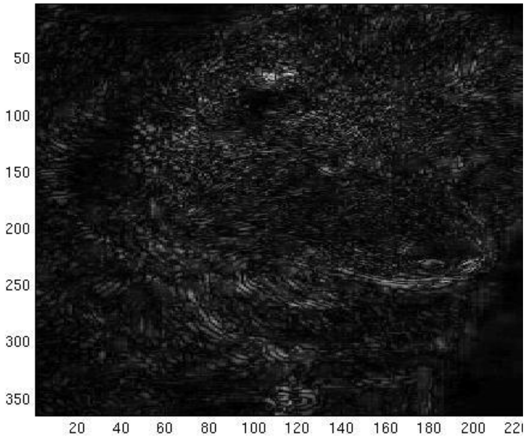
K=1



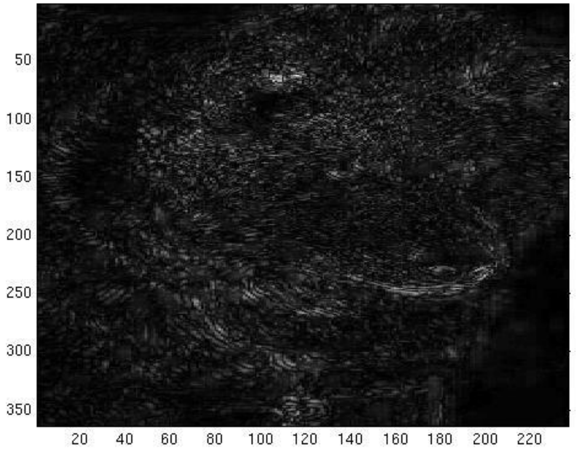
K=2



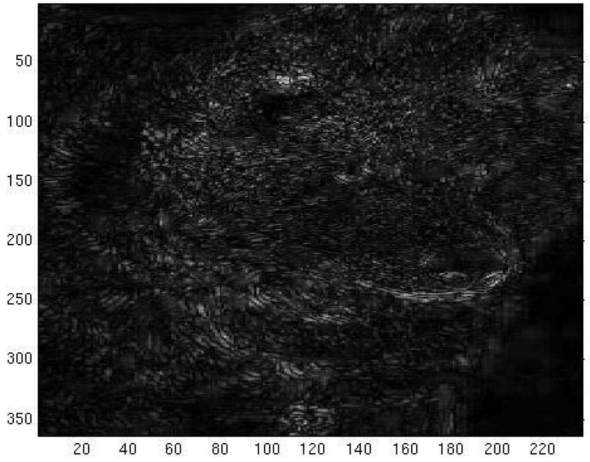
K=3



K=4

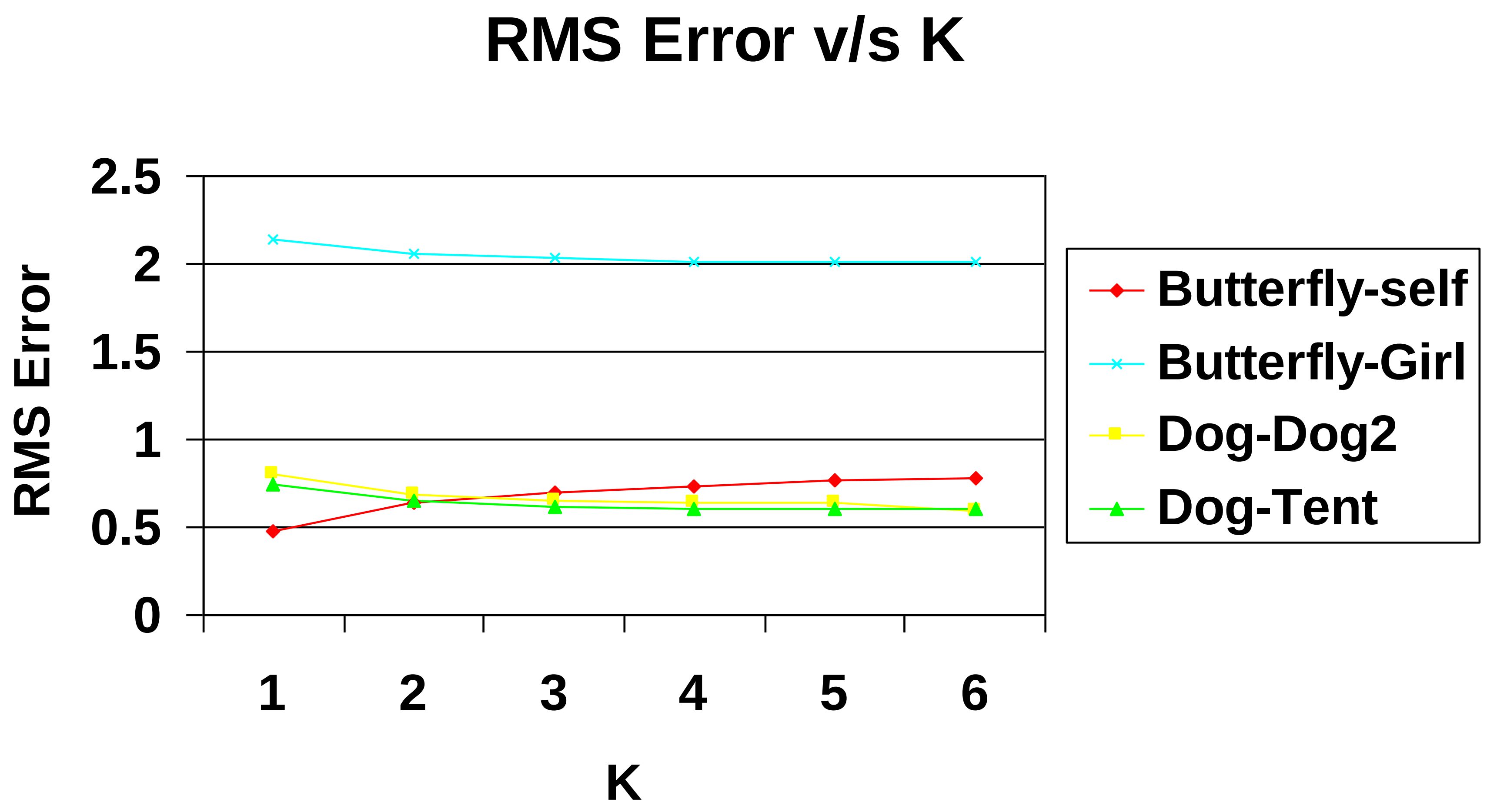


K=5



K=6

Results – *The Graph !*



Comments on results

- Best results when using
  - *Different images*: K = 4 or 5 (confirming to what is stated in the paper).
  - *Same images*: K =1 (why ?))

上海科技大学
ShanghaiTech University

Comments

- Method worked well with our test data.
- Care should be taken about image patches which do NOT form a manifold.
- Limitations on magnification.
- Size/ Overlap in the patch.
- Many extensions are possible:
  - Diagonal gradients for LR (increased dimensionality)
  - Using Lumina as feature vector in LR
  - Can be extended to Video shot SR
  - Multiple training images