Patch Based Blind Image Super Resolution

Qiang Wang, Xiaoou Tang, Harry Shum
Microsoft Research Asia, Beijing 100080, P.R. China
{qiangwa,xitang,hshum@microsoft.com}

Abstract

In this paper, a novel method for learning based image super resolution (SR) is presented. The basic idea is to bridge the gap between a set of low resolution (LR) images and the corresponding high resolution (HR) image using both the SR reconstruction constraint and a patch based image synthesis constraint in a general probabilistic framework. We show that in this framework, the estimation of the LR image formation parameters is straightforward. The whole framework is implemented via an annealed Gibbs sampling method. Experiments on SR on both single image and image sequence input show that the proposed method provides an automatic and stable way to compute super-resolution and the achieved result is encouraging for both synthetic and real LR images.

1. Introduction

The task of super resolution (SR) is to infer a high resolution image from low resolution images of the same scene. One basic idea is that if there are several low resolution images available with sub-pixel displacement, then the high frequency information of the super resolution image can be inferred [10]. However, mathematically, the ML (maximum likelihood) estimation of the super resolution image is highly ill-posed. Various methods are thus proposed to achieve a MAP (maximum a-posterior) solution by using a prior distribution over the high resolution image space.

There are mainly two categories of super resolution algorithms. The first one includes the reconstruction based super resolution algorithms which use the SR image reconstruction constraint together with a generic smoothness prior such as Huber MRF (Markov Random Field) [13] and Bilateral Total Variation [6]. However, the reconstruction constraints provide less and less useful information as the image magnification factor increases [1] [11] and it has often been assumed that the parameter of the camera's point spread function (PSF) is known in advance, which may restrict its usage in various real scenarios. The second group of algorithms are learning based super resolution algorithms which use a learned co-occurrence prior to encode the correspondences between high resolution and low resolution image patches or coefficients of alternative representations [3] [7] [14]. Freeman et al. [7] propose an example-based learning scheme that was applied to generic images where the low to high resolution patch model is learned via a Markov random field and loopy belief propagation is used for inference. The method is somewhat dependent on the training set. So the result is not stable and sometimes produces artifacts in real applications [2].

The models proposed in [1] [5] try to combine the learned patch co-occurrence model and the SR reconstruction constraint in a simple sequential way. Dedeoglu et al. [5] assume that there is a unique high resolution (HR) template whose parameter peaks are around the true high-resolution solution. So it first finds a most likely HR template from the learned patch model through Iterated Conditional Modes (ICM), then the final HR image is estimated from the template in a SR reconstruction framework through gradient based optimization. The method performs well for high zoom super resolution of a very specific class of objects, human faces. In particular, the training face and testing face are from the same person, thus making the super resolution task much simpler. This sequential optimization procedure may have difficulties in finding an optimum solution for general images other than faces. Pickup et al. [12] formulate the two constraints in a quadratic form objective function. However, they found difficulty in calculating the gradient with respect to the texture prior term and simply ignored it. In addition, both [5] and [12] assume that the PSF parameter is known in advance, which we know is difficult to obtain in many real applications. Tipping et al. [15] proposed a Bayesian image SR approach which can simultaneously recover the HR image, PSF parameter and image registration parameters. However, the dimension of the covariance matrix involved in the algorithm is the size of the image, which can be extremely large. In addition, the estimation of the PSF and registration parameters directly from LR images lacks robustness for low quality real images.

In this paper, we propose a new probabilistic framework to integrate the SR reconstruction constraint and the patch

Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05)
1550-5499/05 $20.00 © 2005 **IEEE**

based image co-occurrence prior for image super resolution where the PSF parameter of the real image formation process is unknown (so called blind SR). The integration is not straightforward mainly because of the PSF blurring effect in the reconstruction constraint: a local LR image patch is generated by a set of neighboring HR image patches rather than a single HR image patch as assumed in traditional learning based SR algorithm. We thus formulate the problem in a general graphical model and solve it via annealed Gibbs sampling. The main contributions of this paper include: 1) a general probabilistic framework which combines the SR reconstruction constraint and an image co-occurrence prior; 2) estimation of the PSF parameter in the image formation process; 3) an adaptive locality sensitive hashing algorithm for similarity search in a large training data set for fast implementation of the SR algorithm.

2. Overview

In this section we give an overview of previous works and the proposed method. Fig. 1 shows the graphical model for the SR process. Fig. 1(a) is the MRF network for learning based SR [7]: each HR image patch X_i and the observed LR image patch Y_i are nodes in a Markov network, where the links between nodes indicate statistical dependencies. So the MRF graphical model in Fig. 1(a) implies two things: 1) Knowing the HR patch X_i provides all the information about the observed LR patch Y_i since X_i has the only link to Y_i; 2) The HR patch X_i gives information about nearby HR patches, by the links from X_i to nearby HR patches. By using a discrete representation of MRF nodes, the possible candidate states of HR patch X_i are extracted from the training data by its observed LR patch Y_i, and then the Loopy Belief Propagation algorithm is used as an inference algorithm to obtain the final HR image x.

Fig. 1(b) shows the graphical model for the proposed combination model which better illustrates the real image formation process. It introduces a new node $\mathbf{r}$ representing the PSF parameter of the reconstruction based SR method, which can affect the statistical dependencies between the HR image patch and LR image patches: now the PSF blurring effect provides some additional links between HR patch X_i and the neighboring nodes of LR patch Y_i, which also means that now the observed LR patch Y_i depends both on the PSF parameter node $\mathbf{r}$ and a set of neighboring nodes of HR patch X_i. Note that this graphical model is not a typical MRF model. This paper starts from this graphical model and in the following sections develops an efficient algorithm to jointly infer both HR image x and the PSF parameter r.

3. The probabilistic combination model

In this section, we will first derive the formulations of the proposed combination model and then we will introduce the inference algorithm to get the MAP estimation of the model

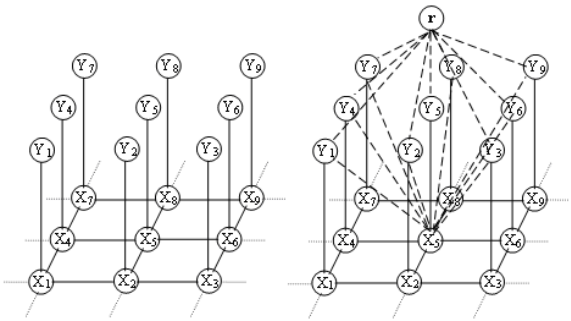

Figure 1. *Graphical model for SR. Left(a): The MRF model [7]. X_i is a HR image patch. Y_i is an observed LR patch. The links indicate statistical dependencies between nodes. Right(b): The proposed combination model. $\mathbf{r}$ is the node representing the PSF parameter. Note that only the links for node X_5 are complete, other nodes' links are omitted for clarity.*

state. The detailed information for learning and implementation will be described in Sec.4 and 5 respectively.

3.1. Formulations of the model

Denote the low resolution input images by $y^{(k)}$, $k = 1...K$, and the HR image to be estimated as x. As shown in Fig. 1(b), the HR image x is represented as $x = \{x_s | s \in S\}$, where x_s is a HR patch indexed by the site s and S is the set of all sites, $S = \{s_1, s_2, ...s_M\} = \{(i,j)|1 \le i \le m, 1 \le j \le n\}$. Similarly, the LR images' patch representation is denoted as $y^{(k)} = \{y_s^{(k)} | s \in S\}$, $k = 1...K$. By taking the PSF parameter r as a random variable and assuming statistical independence between x and r, we get the full posterior distribution of the HR image and the PSF parameter given the model in Fig.1(b):

$$P(x, r|\{y^{(k)}\}) \propto P(\{y^{(k)}\}|x, r)P(x)P(r), \qquad (1)$$

Where $P(\{y^{(k)}\}|x, r)$ can be derived from the reconstruction based image formation model [13]:

$$P(\{y^{(k)}\}|x, r) = \prod_{k=1}^{K} (\frac{1}{2\pi\sigma_k^2})^{N_1/2} e^{-\sum_{k=1}^{K} \|y^{(k)} - W(r)^{(k)} x\|^2 / 2\sigma_k^2} \qquad (2)$$

$W(r)^{(k)}$ is a matrix which models the down sampling, PSF blur and geometric warping effect during the generation of $y^{(k)}$ from x. σ_k is the standard deviation of the independent additive Gaussian noise in the LR image observation. N_1 is the size of the low resolution image. The HR image prior distribution $P(x)$ can be derived from the learning based SR algorithm [7]:

$$P(x) = P(\{x_s\}) = \prod_{(i,j)} \Psi(x_i, x_j), \qquad (3)$$

$\Psi(\cdot)$ is the pairwise compatibility function defined on the neighboring nodes x_i and x_j. Denoting x_i^l, x_j^m as a specific candidate state of x_i and x_j respectively, we have:

$$\Psi(x_i^l, x_j^m) \propto \exp\left(-\|d_{ji}^l - d_{ij}^m\|/2\sigma_o^2\right), \tag{4}$$

d_{ji}^l is a vector of pixels from patch x_i^l which lie in the overlapped region with node x_j; d_{ij}^m is a vector of pixels from patch x_j^m which lie in the overlapped region with node x_i. σ_o is the standard deviation of a Gaussian noise in training patches. The PSF parameter's prior distribution $P(r)$ is assumed to be a uniform distribution over a reasonable range $[r_0, r_1]$ since we have no prior knowledge on it. So we have:

$$P(r) = \begin{cases} 0 & r < r_0, r > r_1 \\ \frac{1}{r_1 - r_0} & r_0 \le r \le r_1 \end{cases} \tag{5}$$

Now we have all the distributions in Eq.(1) which form the posterior of the variables to be inferred. But note that now the joint random vector (x, r) does not form a typical MRF model as indicated in Fig. 1(b) and the observation likelihood in Eq.(2) cannot be easily factorized to represent the observation compatibility functions defined on pairs of HR and LR nodes. Since there are too many compatibility links between the HR node and the observed LR node, it is difficult to model them in an MRF framework as in [7]. As a result, the Loopy Belief Propagation algorithm cannot be used to infer (x, r) from Eq.(2). So we will develop an appropriate inference algorithm in the following section.

3.2. Formulations of the model

The patch-based form of Eq. (1) can be written as:

$$P(\{x_s\}, r|\{y_s^{(k)}\}) \propto P(\{y_s^{(k)}\}|\{x_s\}, r)P(\{x_s\})P(r) \tag{6}$$

The conditional distribution of the above posterior is specified by:

$$P(x_t|\{y_s^{(k)}\}, r, \{x_s : s \neq t\}) \propto P(\{y^{(k)}\}|x, r)P(x_t) \tag{7}$$

$$P(r|\{x_s\}, \{y_s^{(k)}\}) \propto P(\{y^{(k)}\}|x, r)P(r) \tag{8}$$

The first conditional is the posterior distribution of patch x_t given all the other HR patches, LR patches and the PSF parameter. The second conditional is the posterior distribution of the PSF parameter given the HR and LR image correspondences. The basic idea is that if we can efficiently sample from the above two conditionals, then we will be able to get the MAP estimation of $\{x_s\}$ and r through Gibbs sampling [8].

The formulation of the conditional distribution in Eq.(7) will be derived in Sec 3.3 and that of Eq.(8) will be derived in Sec 3.4. After that, the annealed Gibbs sampling algorithm can then be implemented to get the MAP estimation of x, r from Eq. (6), and the implementation details will be given in Sec 5.

3.3. Conditional distribution for HR patch

Denote $\eta_t \subset S$ as the set of neighborhood sites for site t where HR patch x_t is located. From Eq.(3), we have:

$$P(x_t) = \prod_{i \in \eta_t} \Psi(x_i, x_t), \tag{9}$$

Furthermore, the likelihood distribution $P(\{y^{(k)}\}|x, r)$ in Eq.(7) can be significantly simplified because the varying patch x_t only affects a small portion of the whole image and the part that is constant with respect to x_t can be ignored. Firstly, denote $h_t \subset S$ as the set of sites that will affect the observed images at site t in the image formation process. Note that η_t and h_t are different; for example in Fig. 1(b), η_t contains 4 nearest neighbors around site t while h_t contains 9 sites, including 8 nearest neighbors around site t and t itself. Normally, the size of h_t depends on the camera's PSF parameter and the sizes of the HR and LR patch. Secondly, we denote the reconstruction residue image as:

$$\varphi^{(k)} = y^{(k)} - W(r)^{(k)}\widehat{x}. \tag{10}$$

It is the residue image between the k-th observed LR image and the synthesis of the k-th LR image using the current estimate of HR image $\widehat{x}$. Similarly as the patch representation of HR image x, the residue image's patch representation is denoted as $\varphi^{(k)} = \{\varphi_s^{(k)} | s \in S\}$ and Eq. (2) becomes:

$$P(\{y^{(k)}\}|x, r) \propto \exp\left(-\sum_{q \in S}\sum_{k=1}^{K} \|\varphi_q^{(k)}\|^2/2\sigma_k^2\right), \tag{11}$$

Finally, from (7), (9) and (11), we get the conditional distribution of $x_t, t \in S$ as:

$$P(x_t|\{y_s^{(k)}\}, r, \{x_s : s \neq t\}) \propto \prod_{i \in \eta_t} \Psi(x_i, x_t) \exp\left(-\sum_{q \in h_t}\sum_{k=1}^{K} \|\varphi_q^{(k)}\|^2/2\sigma_k^2\right) \tag{12}$$

Such a distribution can be in the form of a discrete distribution if x_t has a discrete state space (the set of candidate patches for x_t) as in [7]. The details of learning the discrete state space of the HR image x will be described in Sec. 4.

3.4. Conditional distribution for PSF parameter

By merging the distribution $P(\{y^{(k)}\}|x, r)$ and $P(r)$, we can easily get the conditional distribution of PSF parameters:

$$P(r|\{x_s\}, \{y^{(k)}\}) \propto \begin{cases} 0 & r < r_0, r > r_1 \\ P(\{y^{(k)}\}|x, r) & r_0 \le r \le r_1 \end{cases}, \tag{13}$$

where $P(\{y^{(k)}\}|x, r)$ is specified in (2). It is difficult to directly sample r from the continuous distribution

$P(r|x, \{y^{(k)}\})$ since it has a very complex form with respect to r. We thus use an importance sampling method instead to approximate the distribution of $P(r|x, \{y^{(k)}\})$ and then sample r through importance re-sampling, the details of which can be found in Sec. 5.

4. Learning patch-based image configuration

Now we have derived all the probability models for our framework. There is still one implementation issue: how to get the discrete state space (the set of candidate patches for each patch in x) of the HR image x. In this section, we will describe how to learn such a patch-based image configuration. The idea to learn an image co-occurrence model similar to that described in [7], and we will briefly describe it in Sec. 4.1. Then we propose a fast similarity search algorithm to deal with the learning problem on a large training data set in Sec. 4.2.

4.1. Image co-occurrence model

The image co-occurrence model models the co-occurrence effect between the patch from a HR image and the corresponding patch from a LR image. The image co-occurrence model is learned from training data. The training data is formed by a set of image pairs. Each pair contains a high-resolution image and the corresponding degraded low-resolution image. The patch pairs are then extracted from the image pairs (with overlap). For both compression and generalization, we use principal components analysis (PCA) to find a set of lower dimensional basis functions for the HR patches and the LR patches respectively. The PCA transformation process can be represented as:

$$\theta = \Lambda_t(\widetilde{\mathbf{d}} - \mathbf{u}), \tag{14}$$

where $\widetilde{\mathbf{d}}$ is the normalized image patch in vector form. $\widetilde{\mathbf{d}} = \mathbf{d}/e$, where e is the energy of the patch, which is the average of absolute pixel intensities. $\mathbf{u}$ is the mean of all the normalized image patches. Λ_t is the PCA transformation matrix and θ is the transformed coefficient of the patch.

Now given an observed LR patch η, we first normalize and get its PCA coefficient. Then we can find an appropriate number of HR patch candidates from the image co-occurrence model by searching the K-nearest neighbors (KNN) of the observed LR patch. Then the PCA coefficients of the found HR patch candidates are used to reconstruct the HR patch. Since a naive KNN search in a large data set is unacceptably slow, we propose a fast KNN search algorithm in Sec. 4.2. Denote one of the candidate HR patch coefficient as θ', then the formulation to reconstruct the corresponding HR patch is:

$$\bar{\omega} = e(\Lambda_d\theta' + \mathbf{u}), \tag{15}$$

where $\bar{\omega}$ is the reconstructed HR patch, Λ_d is the PCA deformation matrix, e is the patch energy of $\bar{\omega}$ to be estimated. Suppose that e_0 is the energy of the observed LR patch η corresponding to patch $\bar{\omega}$. Normally, $e \neq e_0$ since the camera PSF blur effect will change the energy distribution of each patch. We thus turn to an iterative update method by setting the initial value of e to e_0 and then update it by the ratio between the two energies from the currently estimated HR patch and its blurred correspondent patch.

Thus all the candidate HR patches obtained from the image co-occurrence model constitute the state space of the HR image x.

4.2. Fast Similarity search

Since the total number of image patches can be very large (>15 million in our experiments), it is necessary to develop an efficient similarity search algorithm for the KNN search required in Sec. 4.1.

We start from a recently developed algorithm for fast approximate neighbor search, Locality Sensitive Hashing (LSH) [9]. The algorithm indexes the training examples by a number of hash tables such that the probability of collision is large for similar examples and small for dissimilar ones. The LSH algorithm aims at solving the (R, ε)-NN problem. The goal is to report a point within distance $(1+\varepsilon)R$ from a query u, if there is a data point v in the data set X within distance R from u. Otherwise, the absence of such points is reported. A family H of hash functions is called $(R, R(1+\varepsilon), p_1, p_2)$-sensitive if for any $u, v \in X$:
1) If $d(u, v) \leq R$, then $P_{R_H}(h(u) = h(v)) \geq p_1$
2) If $d(u, v) > (1+\varepsilon)R$, then $P_{R_H}(h(u) = h(v)) \leq p_2$,
where P_{R_H} is the probability with respect to a random choice of $h \in H$. We choose the hash family based on 2-stable distributions [4]. Formally, each hash function $h_{\mathbf{a},b}(v) : R^d \rightarrow N$ maps a d-dimensional vector v onto the set of integers. Each hash function in the family is indexed by a choice of random $\mathbf{a}$ and b, where $\mathbf{a}$ is a d-dimensional vector with entries chosen independently from a 2-stable distribution (the Gaussian distribution is 2-stable) and b is a real number chosen uniformly from the range $[0, w]$, where w is the width of projection and the hash function is:

$$h_{\mathbf{a},b}(v) = \lfloor \frac{\mathbf{a} \cdot v + b}{w} \rfloor \tag{16}$$

The problem now is how to choose w. If w is too small, the risk increases that no neighbor of a query is found. If w is too large, then the risk increases that too many neighbors will be found and cause the buckets of the hash table to overflow. Furthermore, the returned result may not be 'similar' enough. Actually, since the density of the training image patches varies, it is not possible to choose a single suitable w.

We thus propose to use an Adaptive LSH (ALSH) scheme. The basic idea is to build a hierarchy of hash tables with respect to different w in the index construction

stage and use adaptive search in the nearest neighbor query stage. Since the algorithm uses variable length bucket numbers during hash table construction, the space complexity of the ALSH is the same as LSH and the computational complexity is at most m times that of LSH where m is the number of $w's$ selected (in practice, the computational time may increase only 1$\sim$2 times). The detailed algorithm for index construction and query is in Fig. 2.

Denote the point set as $\mathbf{P} = \{\mathbf{p_1}, ..., \mathbf{p_n}\}$ and the query point as $\mathbf{q}$. The index construction stage will generate a sequence of projection width $W = \{w_1, ...w_m\}$ and l hash tables $\{T_{i1}, ...T_{il}\}$ for each projection width w_i. The query stage will access the hash tables and generate K (or few) approximate nearest neighbors of $\mathbf{q}$. The algorithm is described as follows:
Index construction:
For $i = 1...m$
1. Set $w_i = 2^i * w_0$
2. Set the number of hash table buckets: $b_i = 2^{-i} * b_0$
3. For $j = 1...l$
3.1 For $s = 1...n$
3.1.1 Calculate k-dimensional projection index $(h_1(\mathbf{p_s}), ...h_k(\mathbf{p_s}))$ for point $\mathbf{p_s}$ according to (19).
3.1.2 Re-hash the projection vector using a standard hash function Y_h. Y_h depends on the buckets number b_i
3.1.3 Store point $\mathbf{p_s}$ on the bucket $Y_h(\mathbf{p_s})$ of hash table T_{ij}
Query:
$S \leftarrow NULL$
1. For $i = 1...m$
1.1 For $j = 1...l$
1.1.1 Calculate the projection vector and its re-hashed value of point $\mathbf{q}$ same as the procedure in step 3.1.1 and 3.1.2 in index construction stage
1.1.2 $S \leftarrow S \cup$ {points found in bucket of $Y_h(\mathbf{q})$ of hash table T_{ij}}
1.1.3 If $|S| \geq n_0$, break.
2. Sort S in increasing order according to its distance

Figure 2. *ALSH algorithm.*

5. MAP estimation

This section describes the algorithm implementation to get the MAP estimation of x, r of the posterior distribution in Eq. (6). The MAP estimation of x and r are carried out using a Gibbs sampler with simulated annealing. Denote all the random variables as $Q = \{\{x_t : t \in S\}, r\}$. In order to perform the simulated annealing, the energy term (negative log function) of all the distribution is divided by a temperature T. The Gibbs sampler starts with an initial temperature T_0 and then gets the sample of Q by flipping it to the candidate state according to the conditional distribution specified in (12) and (13). Then the temperature is decreased iteratively until it approaches 0 and finally the samples of Q will converge to the MAP solution. The algorithm is described in Fig. 3.

Initialization: Set x to a random state in Ω and sampling r from a uniform distribution over $[r_0, r_1]$, Ω is the state space of HR image.
Loop i from 1 to I
1. Set $T = C/ln(1 + g * i)$
2. Loop t from s_0 to s_M: (where $S = \{s_0, ...s_M\}$)
2.1 Calculate the discrete conditional distribution of x_t in Eq.(12), denote it as $q(x_t)$.
2.2 Set x_t to x_t^i according to the probability $q(x_t = x_t^i)$, where x_t^i is a specific state.
3. Loop j from 1 to J
3.1 Sampling $r^{(j)}$ from a uniform distribution on $[r_0, r_1]$
3.2 Calculate the importance weight $w^{(j)}$ of the sample $r^{(j)}$ according to Eq.(13), when r is set to $r^{(j)}$
4. Set $r = r^{(j)}$ according to a probability proportional to its weight $w^{(j)}$

Figure 3. *Annealed Gibbs Sampling algorithm.*

There are some parameters in the algorithm. C is a constant related with the initial temperature T_0 which can be set to 1. g is a speed up parameter that controls the step size of each iteration and we set it to 10 in the following experiments. I is the number of iterations in the annealing process. J is the number of samples that are used to approximate the conditional distribution of r and it depends on the range of possible r. In the following experiments, $I = 5, J = 50$ and the range of r is set to [0.3, 3.0].

Finally, in order to get a pixel level HR result, we apply an iterative back projection operation [10] using the PSF parameter estimated by the above algorithm for a few iterations, where the parameter of the back projection kernel is set to 0.5 and fixed in all the following experiments.

6. Experiments

In order to test the validity of the presented algorithm, we performed two experiments. The first experiment is super resolution from a single image with significant PSF blur. Both a synthetically down-sampled LR image and the real LR image captured from a mobile phone camera are used. The second experiment is super resolution from image sequences where there exist image blur, noise, and registration error.

The training data of the experiment is collected from six different business cards. The HR images are of size 1200*720 and captured by a SLR digital camera (8 Megapixel), the LR images are obtained by blurring the HR im-

ages with Gaussian PSF of s.t.d=1, 2, 3 respectively. Thus we get 18 training image pairs. All possible 7*7 patch pairs are extracted and indexed by the ALSH algorithm described in Sec. 4.2. The obtained training data is used in all the following experiments.

6.1. SR from a single image

In this experiment, we first get a collection of eight HR images from other business cards to generate the synthetic test LR images. The synthetic LR image is obtained by blurring the HR test image and then down-sampled by a factor of 2. The PSF is a Gaussian function with a standard deviation of 2.5 and the corresponding convolution window size is set to 13*13. The results are shown in Fig. 4. Since the original image is too large, only regions of interest in the image are cropped and shown. The bi-cubic interpolation is similar to the directly sampled LR image because of the significant PSF blur. The IBP (Iterative Back Projection) algorithm is our implementation of the algorithm in [10], where the PSF blurring kernel is set to 1.0, 2.0 and the ground truth value of 2.5 respectively. The MRF-LBP (Markov Random Field - Loopy Belief Propagation) algorithm is our implementation of the algorithm in [7]. In order to make fair comparison, all the learning based algorithms use the same training data. We have not compared our result to recently proposed learning based algorithms such as [5] because their method focuses on the super resolution of a specific class of objects and they use a much larger patch size (16*16) which would be too large for general purpose super resolution. Some parameters of the proposed algorithm are as follows: the number of nearest neighbors during similarity search is set to 100, parameter σ_o in Eq. (4) is set to 1.0 times the number of pixels in the overlapped region, and σ_k in Eq. (12) is set to 2.0 times the number of pixels in the defined region. The initial PSF parameter is generated randomly in an interval of [0.3, 3.0]. The non-optimized algorithm's speed is about 30 minutes for the super resolution of a 600*360 size image to 1200*720 on a P-IV 2.8G computer. The speed is relatively slow because 70% of the total processing time is spent on the similarity search of about 34,000 patches (in the test image) for their 100 neighbors in the training data which is stored on disk.

From Fig. 4, we can see that our algorithm achieves the best result. The IBP algorithm even with the ground truth PSF parameters is slightly worse with some small "ringing" artifacts. More SR results can be found in Fig. 5. Note that there is no face in the training images. Fig. 7(a) shows the estimated PSF parameters compared to the ground truth. Note that the estimated PSF blur is the difference of blurring between the LR and the HR images. It shows that our estimation of the PSF parameter is quite accurate.

In experiments of previous researches, LR images are all down-sampled ones of HR images using the same down

Figure 5. *The SR result of face. From left to right, the top row is up-sampled LR, bi-cubic interpolation, MRF-LBP results; the bottom row is IBP with ground truth PSF 2.5, our result and the original HR image.*

sample algorithm as the one used in the training data. In the next experiment, we test our algorithm on real LR images (NOT from down-sampling) captured by a 0.3 Mega-pixel low quality mobile phone camera. This task is much more challenging because the LR images are different from the training data (where LR image is from down-sampling) in both image appearance and geometric lens distortion. The result is shown in Fig. 6. We can see that our result can still improve the readability of the characters. It shows that the combination framework makes the proposed algorithm much more robust to the learning model errors.

In fact, our approach is not only limited to the SR of business card image. In addition, the performance of the algorithm can still be improved if we use more specific training data geared towards a specific class of LR images [5] [12].

6.2. SR from image sequence

In this experiment we create a sequence of LR images from each test HR image. First we blur the HR image with a PSF function of standard deviation 2.5 and the corresponding convolution window size is set to 13*13. Then we down sample it by a factor of 2 and generate 8 LR images by shifting the image according to a uniformly distributed motion vector in the range of [0, 1] pixel along both the X and Y axis. Finally, we add Gaussian noise to the resulted LR images, where the standard deviation of the noise is set to 5 (the range of gray scale is [0, 255]). The SR results from the eight LR images are shown in Fig. 8. The parameters of the proposed algorithm remain unchanged from that in the first experiment. To apply the image co-occurrence model

in Sec. 4.1, we use the median filtered image of the sequence as the LR image input. The PSF parameter for the IBP algorithm is set to the ground truth value of 2.5. From Fig. 8 we can see that the proposed algorithm achieves a visually better result. Note that due to space limitations, the resulting image is shown in reduced size (less than half of the original size). Fig. 7(b) shows the estimated PSF parameters compared to the ground truth. The estimated PSF parameters are stable for all the test sequences despite significant image noise.

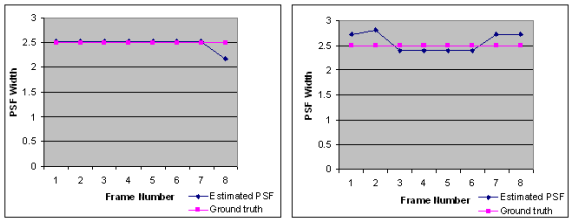

Figure 7. *Comparison of the estimated PSF parameter with the ground truth on Left(a): eight test images (Sec. 6.1) Right(b): eight test sequences with noise (Sec. 6.2).*

Figure 8. *The SR results from eight LR images. From left to right, the top row is direct up-sampling, bicubic interpolation and the IBP result with PSF 1.5. The bottom row is the IBP result with ground truth PSF 2.5, our result and the original HR image.*

7. Conclusions

A method of learning based image super resolution is presented. It combines both the SR reconstruction constraint and the patch based image co-occurrence constraint in a consistent probabilistic framework. The main strength of the proposed method is its ability to estimate the PSF parameter in the SR process and the SR result is better than existing methods. The proposed probabilistic framework is in fact a combination of the global parametric model and the learned local non-parametric model, and we believe that it has the potential to solve other type of low level vision problems.

References

[1] S. Baker and T. Kanade. Limits on Super-Resolution and How to Break Them. *IEEE TPAMI*, 24(9):1167-1183, 2002.

[2] C. M. Bishop, A. Blake, and B. Marthi. Super-resolution enhancement of video. *Proc. Artificial Intelligence and Statistics*, 2003.

[3] D.Capel and A. Zisserman. Super-resolution from multiple views using learnt image models. *Proc. IEEE CVPR*, vol. II, pp. 627-634, 2001.

[4] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. *Proc. twentieth annual symposium on Computational geometry*, pp. 253 - 262, 2004.

[5] G. Dedeoglu, T. Kanade, and J. August. High-Zoom Video Hallucination by Exploiting Spatio-Temporal Regularities. *Proc. IEEE CVPR*, vol. II, pp. 151-158, 2004.

[6] S. Farsiu, M.D. Robinson, M. Elad, and P. Milanfar. Fast and Robust Multiframe Super Resolution. *IEEE TIP*, 13(10):1327-1344, 2004.

[7] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael. Learning Low-Level Vision. *IJCV*, 40(1):25-47, 2000.

[8] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the Bayesian restoration of images. *IEEE TPAMI*, 6(6):721-741, 1984.

[9] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. *Proc. 25th Int. Conf. on Very Large Data Bases*, pp. 518-529, 1999.

[10] M. Irani and S. Peleg. Improving Resolution by Image Registration. *CVGIP*, 53(3):231-239, 1991.

[11] Z. Lin and H.Y. Shum. Fundamental limits of reconstruction-based superresolution algorithms under local translation. *IEEE TPAMI*, 26(1):83- 97, 2004.

[12] L. C. Pickup, S. J. Roberts and A. Zisserman. A Sampled Texture Prior for Image Super-Resolution. *Advances in Neural Information Processing Systems 15*, 2003.

[13] R. Schultz and R. Stevenson. Extraction of high-resolution frames from video sequences. *IEEE TIP*, 5(6):996-1011, 1996.

[14] J. Sun, H. Tao, and H. Shum. Image Hallucination with Primal Sketch Priors. *Proc. IEEE CVPR*, vol. II, pp. 729-736, 2003.

[15] M. E. Tipping, C. M. Bishop. Bayesian Image Super-Resolution. *Advances in Neural Information Processing Systems 14*, 2002.

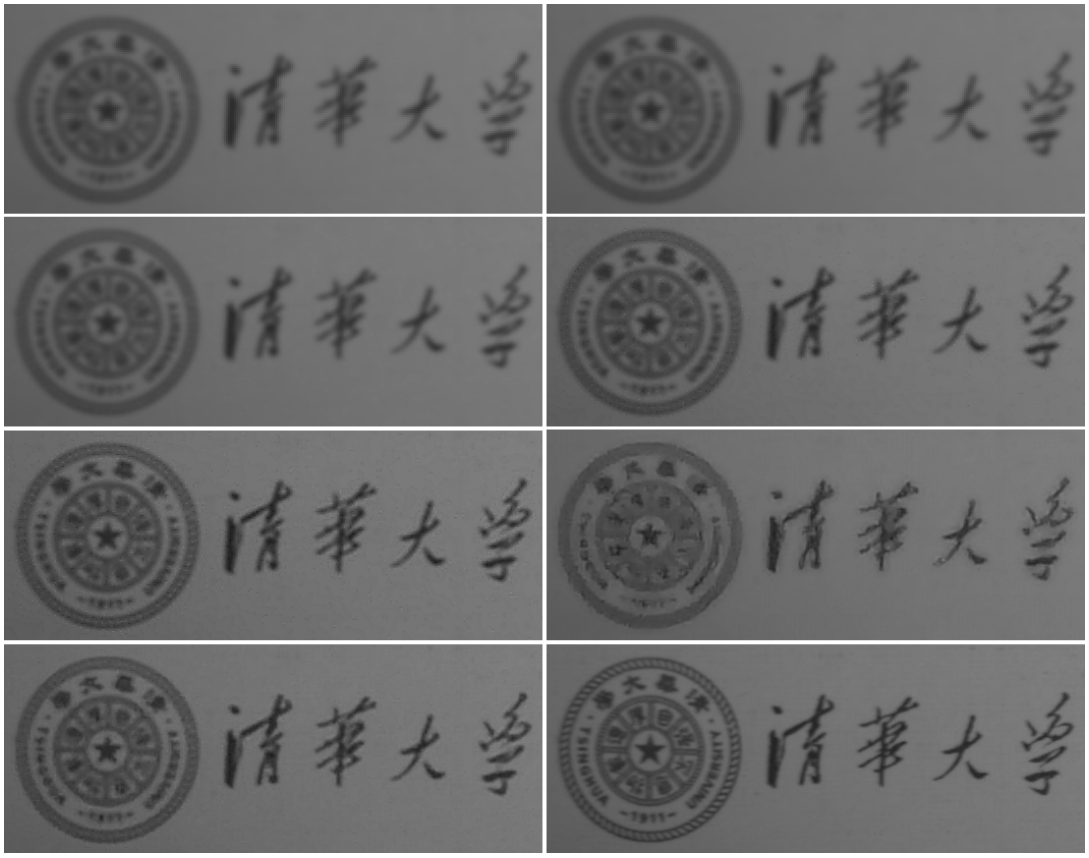

Figure 4. *The SR result of text and logo. From left to right, top row is direct up-sampling and bi-cubic interpolation result, second row is IBP result with PSF 1.0 and 2.0, third row is IBP result with ground truth PSF 2.5 and MRF-LBP result, bottom row is our result and the original HR image.*

Figure 6. *The SR result of a real LR image captured by a mobile phone camera. From left to right, top row is up-sampling and bi-cubic interpolation result, second row is IBP result with PSF 1 and 2, third row is MRF-LBP and our result.*