Final Project Report for B553
Face Hallucination

Chaitanya Bilgikar
School of Informatics and
Computing
Indiana University
Bloomington, Indiana 47405
Email: cbilgika@indiana.edu

Nasheed Moiz
School of Informatics and
Computing
Indiana University
Bloomoington, Indiana 47405
Email: nmoiz@indiana.edu

***Abstract*—In this paper, we have tried to understand and reimplement the face hallucinating or image super resolution process as done by Liu et. al [1].They have used a two step process for this purpose. The first is a global parametric model to account for the global image as a whole, and a local nonparametric model which is used to accurately model and depict the relative of positions of an image of a human face. Their paper uses many techniques from statistics, linear algebra and probabilistic graphical models.**

I. Introduction & Motivation

Our recent learning experience with probabilistic models and our interest in computer vision led us to look for an interesting project that involved both these areas. Image super resolution is one such application at the junction of vision and probabilistic models. But it too, is quite vast as there are many different kinds of super resolution applications and techniques today. A simple application of this concept, as mentioned in the paper, is retrieving high resolution images from low resolution old photos, in which we are particularly interested in the face of one person.

There have been several methods that have been used (as described in the Related Work section) to solve this problem. We chose this one because we felt that it reflected the concepts that we have studied in B-553 better than some of the other papers. The technique used in [1] seems intuitive to a certain extent; they consider the high-resolution image, the output, as a sum of the output of the global and local steps of "face hallucination". The global part tries to model the output image based on statistical models of the training images. The local part then models the statistical similarities from localities on the image using a Markov network. This is done so that the patches are more accurate to the original high quality image relative to each other. This also helps because we are dealing with faces where relative positions of objects like eyes, nose, lips etc are important, as opposed to when we are working with textures and other more general images. In the latter case a small glitch can go unnoticed or unperceived but in case of images of faces, a small glitch draws the viewer's attention drastically to itself and has a greater detrimental effect on the output.

A. Related Work

Face hallucination is a special and unique kind of image super resolution, in that it only deals with images of human faces. As such the results we have seen for face hallucination have been more promising than those of the more general kinds of super resolution. This is most probably because when the image composition is always a face, it is easier to train the model, making it easier to fill in pixels from trained data and provide an accurate, higher resolution image. Before selecting [1] for deeper research and attempting to re implement it, we looked at a few other works which have tried to do Face hallucination using different models and techniques. [2] for instance tackles the problem but without any use of Markov Random Fields. They argued that the Markov assumption is a weak one, albeit one that does aid resolution enhancement. Instead they have used an algorithm which makes use of Gaussian pyramids involving repeated smoothing and down-sampling in small steps. We did not understand the finer details of their hallucination process, but we did figure out that despite not using Markov fields, they were using a MAP formulation, made to fit their own model. [1], too have formulated their process as a MAP assignment, however, as is obvious, their energy function and its numerous components were different from those described by [2]. [3], in a more recent publication criticize both the aforementioned ways of hallucinating faces and instead adopt a method based on sparse coding. They feel that Gaussian pyramids [2] cannot model the facial priors well, that first step in [1] outputs an image that looks too much like the mean image and that their second step involving the patch based Markov network is overly complicated. Out of these three papers examined above,results in [3] were the most successful in replicating the original high resolution image. However, given that we were restricted to pick a method that made use of probabilistic models, we went with [1] as even their results were quite successful. Another very interesting piece of literature that we came across was [4]. Here, they have talked about image super resolution in general, not just with faces. [4] use Markov Random Fields and Loopy Belief propagation transport statistical information long distances from one part of the image to another. Their image manipulation techniques, however, were a little advanced for use to understand.

II. Overview

In this section, we attempt to describe in detail, our understanding of the concepts expressed in [1]. The problem at hand is as follows:

Given a set of high-resolution images and an input low-resolution image, we must compute a high-resolution image such that it satisfies the three main constraints which are:

1. **Sanity Constraint:** The output image must be very similar to the input image after smoothing and down-sampling.
2. **Global Constraint:** The output image must have characteristics that are unique to a human face. In other words, it must have eyes, nose, mouth etc.
3. **Local Constraint:** The output image must have the specific features of the input image with its local features (like the distance between the eyes, nose, lips etc that are unique to the input image).

In terms of mathematics, we can represent the problem as

$$I_H^* = \arg\max_{I_H} p(I_L|I_H)p(I_H) \tag{1}$$

where I_H is the high-resolution image, IL is the low-resolution image that is given as input and I_H^* is the output high-resolution image.
In this paper, the authors have decided to split the image I_H^* into two parts I_H^{g*} and I_H^{l*} which represent the global and local face images respectively. I_H^{g*} makes sure that I_H^* satisfies constraint 2 while the local face image I_H^{l*} satisfies constraint 3. Here it is assumed that the image I_L is A times smaller than I_H.

A. The Global Constraint

In this section, all the training images are considered to be 1-D vectors consisting of N elements where N is the total number of pixels in the high-resolution training image. We then compute the eigenvectors and eigenvalues for this matrix of vectors using PCA analysis. We pick a value l and take only the first l eigenvectors which correspond to the highest l eigenvalues. We form a diagonal matrix Λ from the top l eigenvalues. The mean of all the training images is called μ. We then compute a matrix X^* from the equation given in the paper as:

$$X^* = (B^T A^T AB + \lambda\Lambda^{-1})^{-1} B^T A^T (I_L - A\mu) \tag{2}$$

This X^* is then used to get the best global high-resolution image I_g^* as:

$$I_g^* = BX^* + \mu \tag{3}$$

B. The Local Constraint

Once the image from the first step has been acquired, the second step is performed on it. [1] utilize a Markov Random field where the image is decomposed into patches which are nodes for our Markov network. The image below (fig 1) illustrates the network. The top layer is the image we are trying to infer from the output of the first step, and the bottom layer is the global high resolution image which is all the training images taken one by one. The Gibbs function for this Markov network as defined by [1] is composed of two components the External component and the Internal component. The external component is as follows:

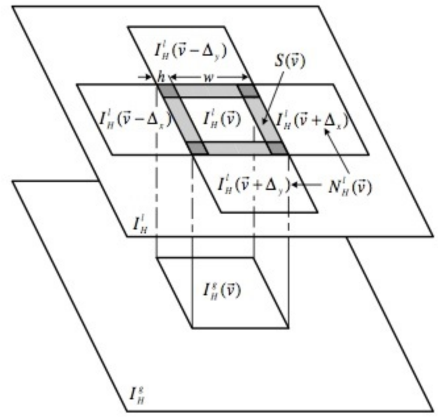

Fig. 1. Markov Network as in [1]

$$E_G^{ext}(\overrightarrow{v}) = \frac{1}{\lambda'} \sum_{i=1}^{k} \delta[I_H^l(\overrightarrow{v}) - I_H^{l(i)}(\overrightarrow{v})] d^2[I_H^g(\overrightarrow{v}), I_H^{g(i)}(\overrightarrow{v})] \tag{4}$$

The λ' in this function is the variance which is a preset scalar. Then for each image in the training set of size K, we take the Dirac of the difference between the patch at position v in the training image i and the patch at the same position in the image from step 1. The product of this multiplication with the norm sum of squared differences of the Laplacian for the two same patches (one from training image i and one from the output of step 1), is one sum in the entire summation. We can get E_G^{ext} by summing over these values for each of the training images.
In the next step, we want to calculate the internal component. In order to this, we take each patch, one at a time, and use this equation:

$$E_G^{int}(\overrightarrow{v}) = \frac{1}{\lambda''} \sum_{\mathbf{u} \in S(\overrightarrow{v})} [I_H^l(\mathbf{u}) - N_H^l(\mathbf{u})]^2 \tag{5}$$

As described in the paper, each patch at position v has a set of neighbors S. The internal function for that patch is the sum of the squared differences between itself and each of its neighbors. Note that in some cases a patch may have only 2 or 3 neighbors only if it is on the edge of the image. Also, the patches themselves overlap their neighbors by a few pixels. This is done so that the resulting final image does not look like a collage with visible edges between patches. Thus, in this model, not only is influence propagated via the Markov network but also because the patches actually have pixels common to them, the assignment to the peripheral pixels for one patch is exactly the same as for the other.
With this step we get the two components of the Gibbs function which need to be added together. Once we have this for each patch, summing over this values of this function for each patch will provide us the total energy for the Markov network. Finally, minimizing this energy using a MAP algorithm such as Max product will output the final result. Note, in addition to what we have understood and explained here, [1] in their implementation have used some other subtle techniques such as simulated annealing which have no understanding of. Additionally, while they loop through the patches they have talked about flipping every patch in the candidate output image, but to what end, we could not exactly understand.

III. IMPLEMENTATION & ISSUES

In this section we aim to describe the approach that we have taken in implementing the methods and concepts described above, along with the challenges that we faced in doing so.

A. Global Constraint

We implemented this part successfully, as can be seen from the code submitted. The code is in MATLAB as it was easy to perform the many matrix operations that were required. The variables have been named as they appear in [1] so as to avoid confusion. The image I_g^* obtained at the end of this stage is dependent of the value of λ chosen. We have observed that a value between 0.2 and 0.3 gives fair results.

1) Issues Encountered:

- The biggest challenge in this paper was understanding the mathematical notations used.There were some parameters that were not very well-defined and hence caused ambiguity during the implementation. For instance, the concept of representing an image as a 1-D long vector was difficult to grasp and even harder to implement.
- PCA as a concept was new to us and we struggled a bit due to the different functions that are available in MATLAB to retrieve the eigenvectors and eigenvalues of a matrix, each having different inputs and outputs (eg: $princomp$ and svd).
- We encountered a problem with MATLAB wherein performing this first step is too slow. We also found that MATLAB tends to run out of memory when more than 20 images are used.

B. Local Constraint

We started implementation of the local constraint part of the face hallucination project but were unable to complete it. This was due to several reasons.

- We both lacked the experience in working with vision techniques and Markov network related projects. Till date, we had only worked with one Markov network and inference based project which was for a significantly simpler Markov network.
- We felt that the Markov Network described in the local constrain part of the paper was overly complicated and involved many terms, and techniques that we were totally unfamiliar with.

However, we did write some code for this part which does not really output anything tangible and is a work in progress. The output that our code currently outputs is only a result of the first step. With that said, the contents of our incomplete attempt to implement the local constraint are described as follows. We initialize a 16x16 matrix of many 8x6 matrices. Here each 8x6 matrix is a patch and hence a 128x96 image has 16x16, 8x6 patches. While we iterate over the training images, for each patch we calculate the external part of the Gibbs function as described earlier in the paper, which comprises of the Dirac function and the difference between the Laplacian of two patches. This is stored in the matrix described above at corresponding indexes. We keep adding to it as we iterate over the training images. We could not get around to implementing the internal component of the Gibbs function as we were short on time and decided to focus on writing the paper.

IV. CONCLUSION

We struggled a lot to understand this paper from the start as it was quite advanced relative to our knowledge and understanding of this field. However, we learned a lot at every step of it. We definitely felt like we bit more than we could chew given the scope and time-frame. Ironically, although this was meant to be an exercise in probabilistic models for us, we could not get around to the part where we implemented the actual Markov networks. Yet, we learned a lot of things from imaging, vision and linear algebra, that often go hand in hand with probabilistic models. Overall, while we did not get satisfactory tangible results, it was a really great satisfying and enriching learning experience for the both of us.

REFERENCES

[1] Ce Liu, Heung-Yeung Shum,Chang-Shui Zhang *A two-step approach to hallucinating faces: global parametric model and local nonparametric model*,Computer Vision and Pattern Recognition, 2001. CVPR 2001. *Proceedings of the 2001 IEEE Computer Society Conference on , vol.1, no., pp.I-192,I-198 vol.1*, 2001

[2] Simon Baker,Takeo Kanade *Hallucinating Faces*, 2000

[3] Jianchao Yang, Hao Tang, Yi Ma, Thomas Huang *Face Hallucination via Sparse Coding*,2008

[4] Bill Freeman and Ce Liu, *Markov Random Fields for Super-resolution and Texture Synthesis*, 2010