

Superresolution using Papoulis-Gerchberg Algorithm

Deepesh Jain

EE392J – Digital Video Processing
Stanford University, Stanford, CA
jaindee@stanford.edu

ABSTRACT

Superresolution, an open-ended research problem still hasn't made its place in lot of textbooks on Video Processing. It still resides in the so-called '*technical paper*' domain. But, hopefully we will soon see Superresolution algorithms being implemented in digital cameras and/or as stand-alone software as they offer a wide variety of quality enhancement for images/video. Superresolution, on the most basic level, deals with construction of a high-resolution image given few low-resolution images with some motion between them. This report examines one of the simple and faster algorithms on Superresolution^{2,3}.

Information theory says – '*Information doesn't hurt*'. So, basically the non-redundant information from the given low-resolution images is used to construct a high-resolution image. Most of the Superresolution algorithms deal with this problem using two steps – Motion estimation between different images and Projection of LR pixel values onto the high-resolution grid. This paper uses Papoulis-Gerchberg^{6,7} algorithm for achieving the projection part and restricts the motion between LR images to a global one – shift & rotation. The algorithm is tested with both synthetic and real images. The results turned out to be extremely positive. Rest of the report discusses the motivation, problem description, proposed solution, technical aspects of the algorithm, results and small discussion on future work. In further discussion LR will be used to refer low-res, HR for high-res and SR for Superresolution.

1. MOTIVATION

A motion video has large number of images in it and Superresolution can be used to extract action-packed images from it, which otherwise are very difficult to capture using a still camera. This can be very useful for sport photographers. They can be used to create movie posters for a movie too. Another use can be to print great gymnastic acrobatic snaps by capturing it as a video first. Scientific applications include astronomy, medical imaging, etc.

Another interesting application would be to produce a high-resolution video from a low-resolution one. If this can be done efficiently, then Superresolution can be used for compression, i.e. transmit only low-res video at the transmitter, and show HR video at the receiver. The applications are unlimited.

The actual motivation behind this project was to get a good intuition on some of the Superresolution algorithms and set path for future research.

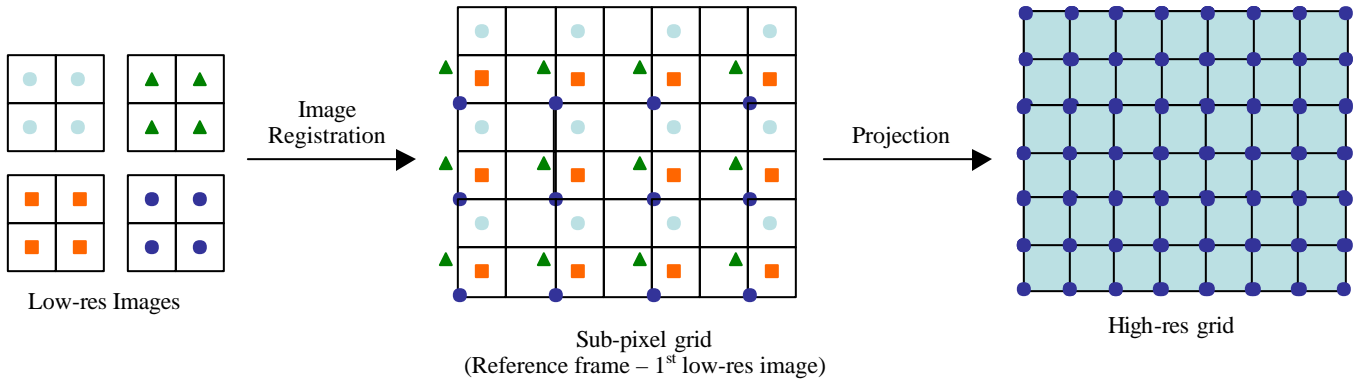
2. PROBLEM

As mentioned earlier, given a bunch of LR images, Superresolution involves two steps:

- Image Registration
- Projecting LR image values onto high-resolution grid

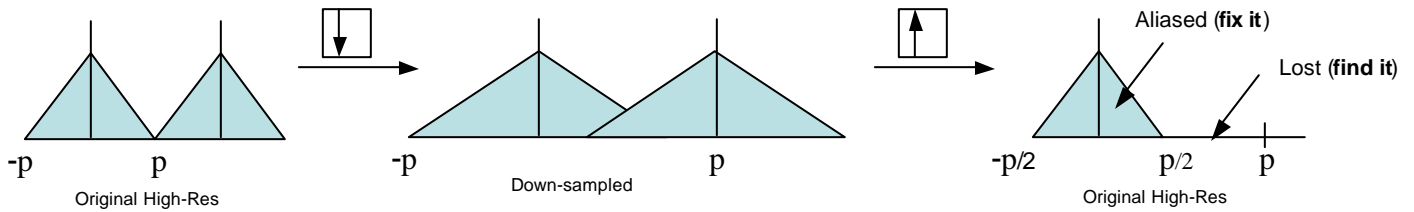
Most of the papers on Superresolution try to solve these two problems. Though their approaches are different but the end goal is same. The following figure describes the two steps graphically. On the left hand side four low-res images are shown (pixel values at integer pixel only). Motion estimation is used to estimate the pixel positions of the three images with respect to the 1st image. Note that these pixel positions would not be at integer values, but can take any real value. Once this information is calculated accurately, it is possible to

project this information on a desired high-resolution grid. Details of how these two problems are solved will be discussed in the following sections.

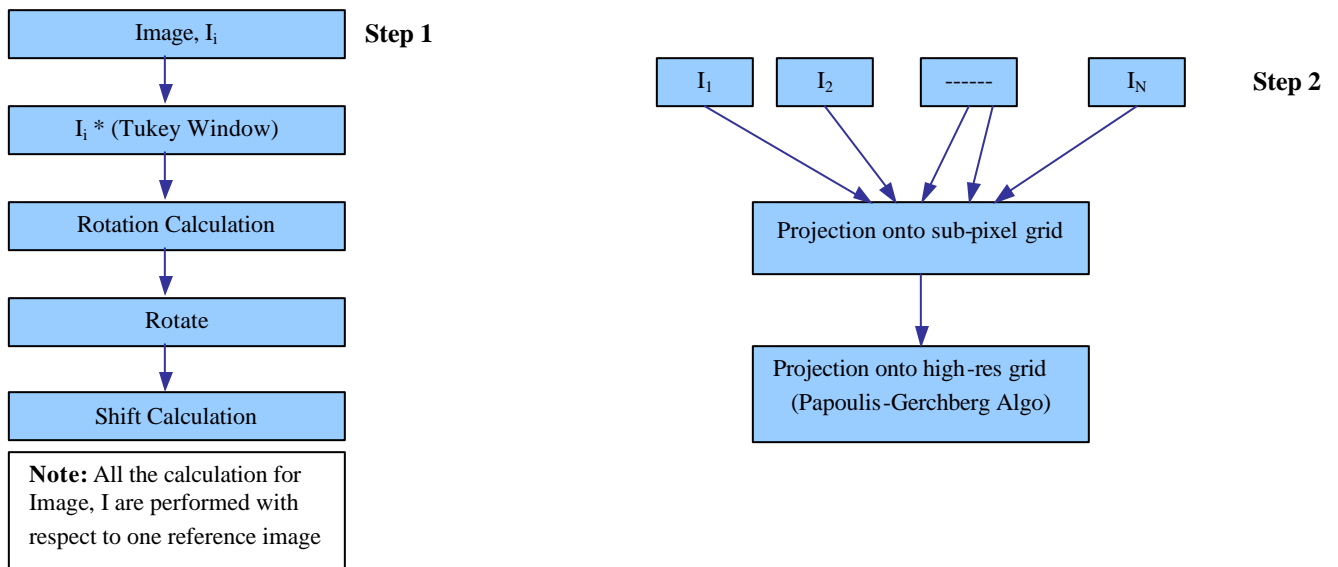


3. SOLUTION

Before delving into the solution, let's first try to understand what we are trying to achieve. Let's say we start with a high-resolution image. Then we downsample it (without low-pass filtering). So, we have aliasing in the down-sampled images. Now, in real-life we will be given these down-sampled images with aliasing and our goal is to create an HR image from them. So, basically as shown in the figure below, it boils down to finding the lost high-freq components and fixing the aliased low-freq values in the HR image. The Papoulis-Gerchberg algorithm (discussed in more detail below) along with POCS is used to solve this problem. It concentrates mainly on fixing the low-freq components, but it turns out that this algorithm also gives us some high-freq components depending on the information in the LR images.



The two steps used to solve this problem are shown below. The details of each step is given in the following sections.



3.1 IMAGE REGISTRATION

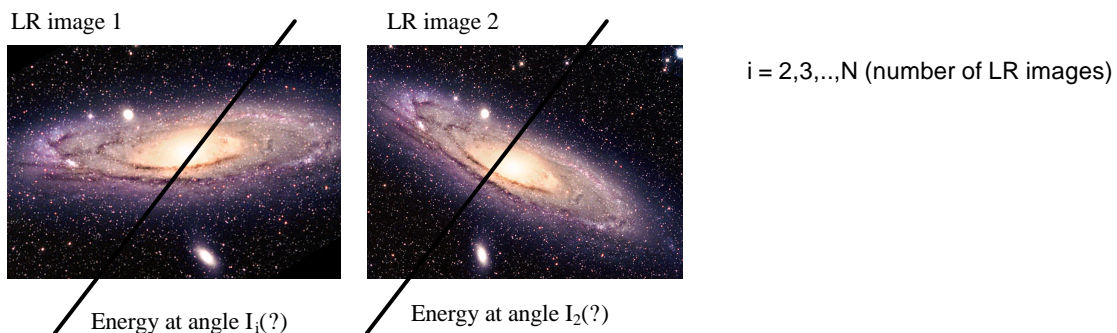
The first step in any Superresolution algorithm is to estimate the motion between given LR frames. A good ME is a hard prerequisite for SR. In this project the motion is restricted to shifts and rotation, so a very simple (though accurate) approach is enough for image registration.

Rotation Calculation

There are two ways that can be used to carry out rotation estimation between low-res images.

- Rotate individual images at all the angles and correlate them with the first image. The angle that gives the maximum correlation is the angle of rotation between them.
- The method described above is computationally expensive, so a faster method is to calculate the image energy at each angle. This will give a vector containing average energy at each angle. The correlation of these energy vectors of different images will give the angle of rotation. The energy calculation could also be carried in frequency domain. A graphical representation below helps to illustrate the method clearly.

$$\text{Angle}_i = \max \text{index} (\text{correlation} (I_1(?), I_i(?)))$$



It turned out that though the first method is computationally expensive, but gives more precise results, so it was used in this project.

Shift Calculation

Once rotation angle is known between different images, shift calculation can be performed. Before calculating the shift, all the images are rotated with respect to the first image. Block matching was not used here for two reasons – 1. It is computationally expensive. 2. We need sub-pixel accuracy that is very hard to get with block matching. A much faster and reliable way to achieve this is using frequency domain method.

$$F_i(\mathbf{u}^T) = e^{j2p\mathbf{u}^T \mathbf{s}} F_1(\mathbf{u}^T) \quad \begin{matrix} \mathbf{s} \rightarrow [\gamma \ x \ \gamma]^T \\ \mathbf{u} \rightarrow [f_x \ f_y] \end{matrix}$$

$$\mathbf{u}^T \mathbf{s} = \frac{\text{angle}(F_i(\mathbf{u}^T) / F_1(\mathbf{u}^T))}{2p}$$

Technically, each frequency bin should give the same shift, but that does not happen. So, a least square method is used to estimate the shift \mathbf{s} . As the LR images could be aliased, only 6% of the lower frequencies were used to calculate \mathbf{s} .

3.2 PROJECTION ONTO HIGH-RES GRID

Once we have a good estimate of shift and rotation between all the images and the reference image, we can find the pixel positions of all the LR images in the reference frame of the first image. Then we can project this information on high-res grid. There are various ways to do this.

- **Bicubic Interpolation³**: Matlab has a function called '*griddata*'. One can pass the non-integer pixel locations and their values to this function and it will return the pixel values at integer pixels. This method was tried but did not give very good results.
- **Iterative back-projection⁴**: This method assumes that each pixel in a low-res image is a result of some integration of pixels in high-res image that are dependent on the camera characteristics. It is very difficult to estimate these projection functions, and so this method tends to be complicated. Also, most of the time camera characteristics are unknown and these parameters have to be tweaked for each set of images.

Taking into account all these considerations, this project used a method that was fairly simple and straightforward - Papoulis-Gerchberg algorithm. It is a special case of POCS (Projection onto convex sets).

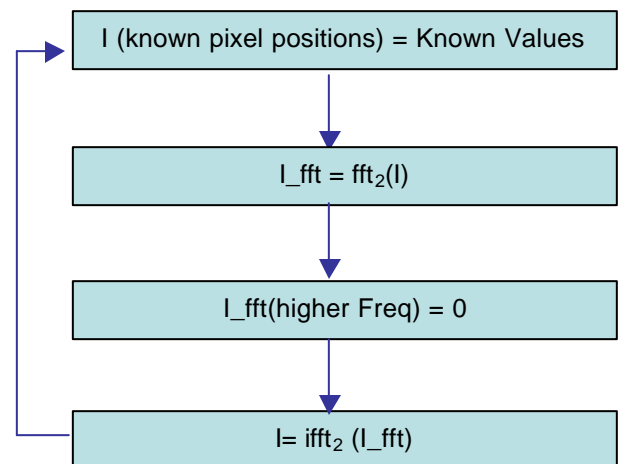
Papoulis-Gerchberg Algorithm^{5,6,7}

This method assumes two things:

- Some of the pixel values in the high-res grid are known.
- The high frequency components in the high-res image are zero.

It works by projecting HR grid data on the two sets described above. The steps are:

- Form a high res grid. Set the known pixels values from the low-res images (after converting their pixel position to the ref frame of first low-res image). The position on the HR grid is calculated by rounding the magnified pixel positions to nearest integer locations.
- Set the high-freq components to zero in the freq domain.
- Force the known pixel values in spatial domain. Iterate.



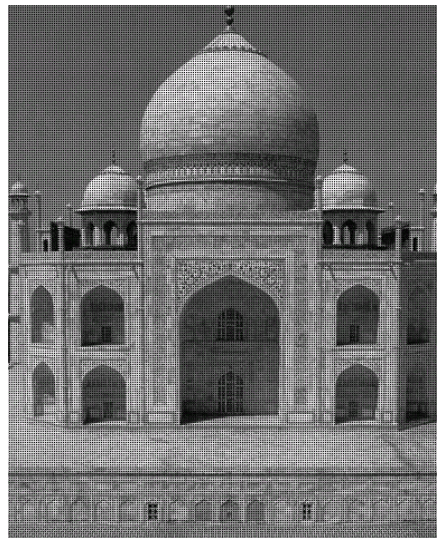
By making the high-freq equal to zero, this method tries to interpolate the unknown values and so correct the aliasing for low-freq components. Also, by forcing the known values, it does predict some of the high-freq values.

The following set of images walks through the actual working of this algorithm. Initially, we fill the HR grid with known pixel values and make the unknown pixel values to be zero. This can be observed in the image shown below. The vertical & horizontal black grid lines represent the unknown pixels. The DFT of this initial estimate shows both high and low-freq components.

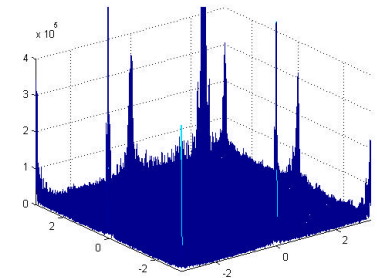
Initial Setup



Taj Mahal – Low-res image I



Initial Image
Contains both known and
unknown pixels (that appear black)



FFT (initial image)

In the next step, the higher frequencies can be made zero in the frequency domain. This effectively is low-pass filtering the image. The center image shows the effect. The unknown pixels now have got some value, and the known values have gone down in amplitude, due to low-pass filtering. We can then increase the magnitude of known pixels by forcing them to what they should be (as can be observed in the rightmost image). This again creates some high-freq components. So, basically by iteratively doing this again and again, we are correcting the low-freq values (by guessing the values for unknown pixels) and finding some the high-freq components by forcing the known values.

Iteration 0 → Iteration 1

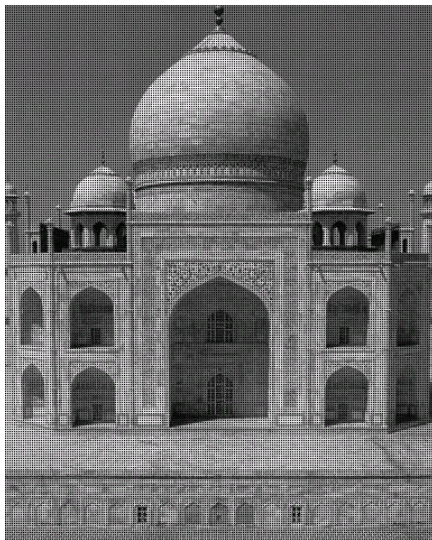
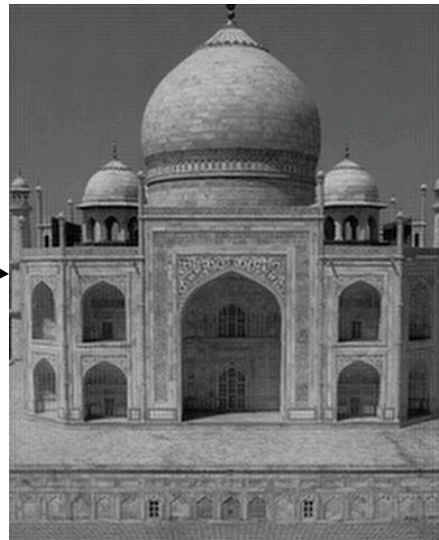


Image at iteration 0



Force
Known
Pixel
Values

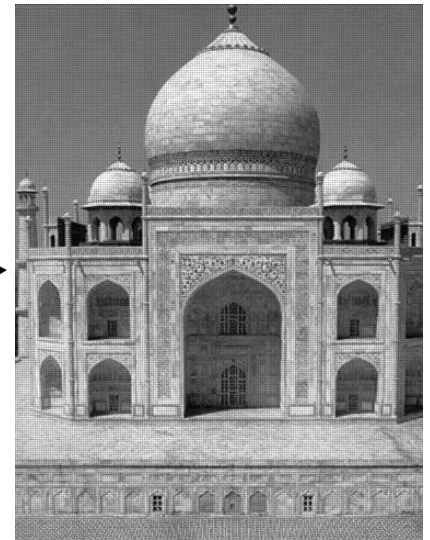
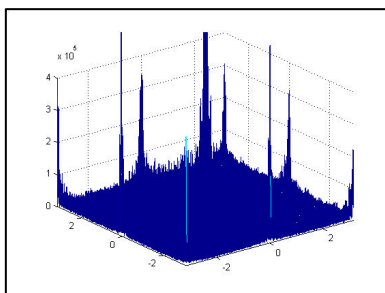
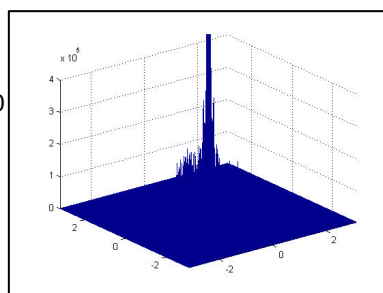


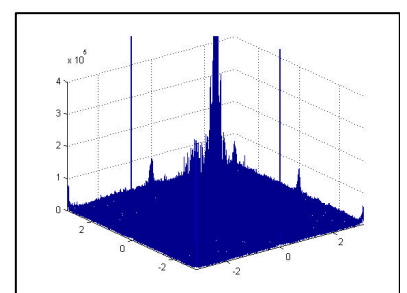
Image after 1st iteration



I (high freq) = 0

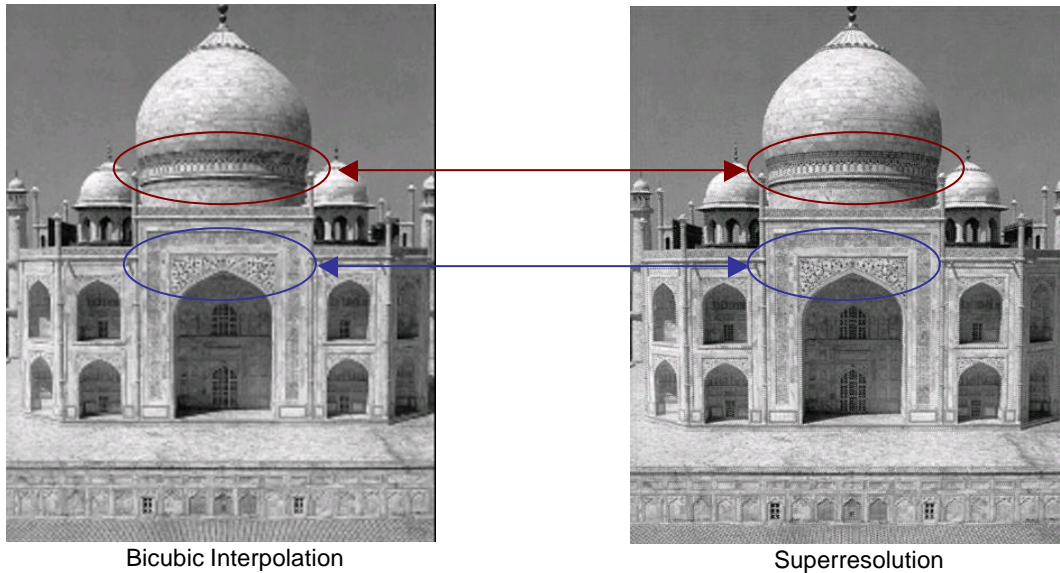


Force
Known
Pixel
Values



After a few iterations, the HR image converges to an SR image, which can be observed below. By juggling between the two data sets, i.e. forcing the high freq to zero and forcing the known values, we have estimated the value of unknown pixels. It turns out that the final SR image looks pretty good. Hence, it can be concluded that Papoulis-Gerchberg algorithm works well for Superresolution.

After 50 iterations



4. RESULTS

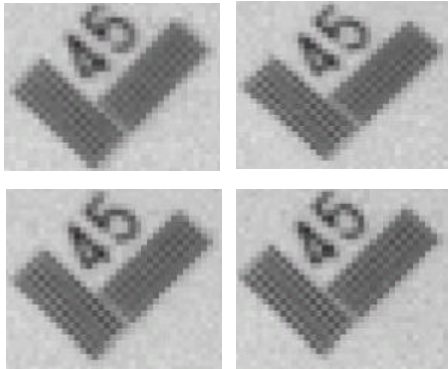
This section describes the experimental setup and the results obtained by using the above algorithm on bunch of images. The PSNR could have been calculated, but all we care about is how the HR looks visually. It was pretty clear from the results that SR had an edge over any conventional method, so calculation of PSNR or MSE would have been nothing more than formality!

Two sets of experiments were carried out.

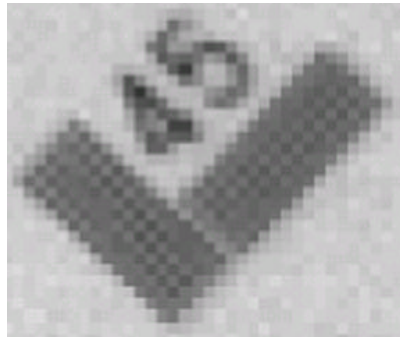
1. Synthetic images – In this experiment, one high-res image was taken, it was randomly shifted to create 4 images. These images were then down-sampled (without low-pass filtering). The four down-sampled images were fed in the algorithm. Three different experiments were performed to test not only the common case but also boundary cases. Common case includes when the various low-res images have some redundant and some non-redundant information among them. Boundary cases involve low-res images that either have only redundant or no redundant information among them.
2. Real images – Four images were taken with the camera and SR algorithm was applied to create one HR image.

4.1 RESULTS – REAL IMAGES

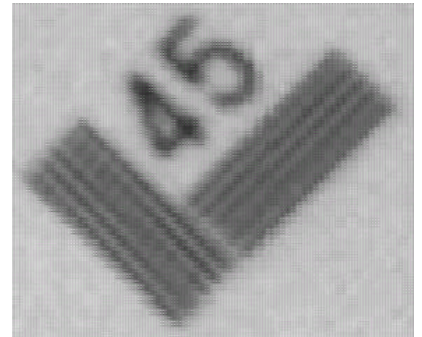
The following low-res images were obtained from the paper by Vandewalle, Susstrunk & Vetterli². As can be observed, there is a big difference in the results of bicubic and Superresolution. The SR algorithm exploited the fact that low-res images have non-redundant information and results in a much sharper looking image.



Original Low-res images



Bicubic Interpolation (1st image)

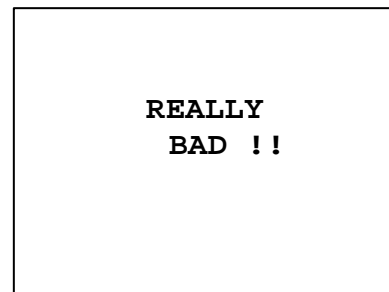
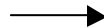


Superresolution Result

In the second experiment, a few images of *Stanford Quad* were taken and Superresolution algorithm was applied on them. It turned out that the final image looked pretty bad. On close observation it was found that the images had motion other than just shift and rotation - affine motion. So, this observation can be used to conclude that correct image registration is essential to Superresolution. Infact it is very boolean, i.e. **if you have a correct registration Superresolution performs well, but if you don't have good motion estimation then Superresolution fails.**



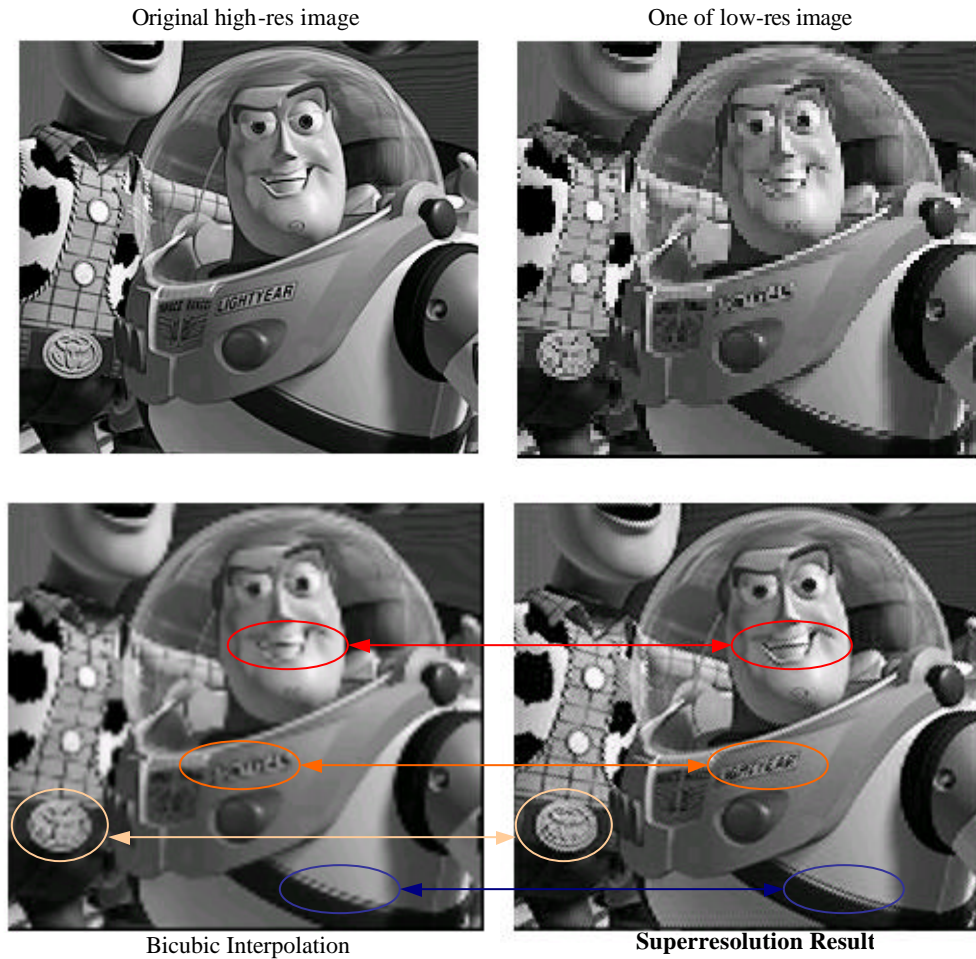
Original low-res images



Superresolution Result

4.2 RESULTS – SYNTHETIC IMAGES

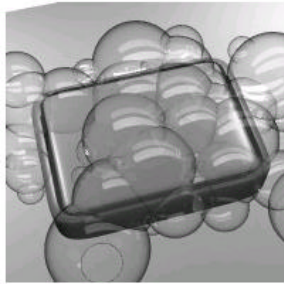
The first set of low-res images were created by randomly shifting the high-res images and then down-sampling them. This is the most common case in real-life where the shifts in low-res images are random and so they have some redundant and some non-redundant information among them. As can be seen clearly, Superresolution gives very good results.



The second set of images represent a boundary condition, where all the low-res images are shifted such that they have the same information, which is analogous to saying that they are moving at critical velocities. As can be seen, Superresolution does not work very well here.



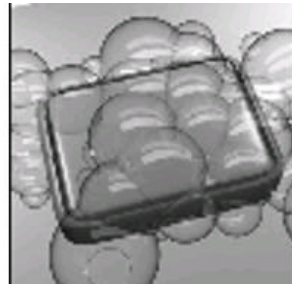
The third set of images represent the other side of boundary condition where the high-res image can be perfectly constructed if the low-res images have all the information needed, i.e. they have no redundant data among them. In this case, the low-pass filtering had no effect, as all the pixel values were known.



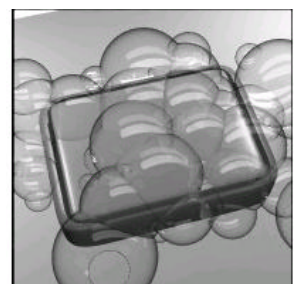
Original high-res image



One of low-res image



Bicubic Interpolation



Superresolution result

5. CONCLUSION

It was observed that Superresolution did give an edge over bicubic interpolation, and can be used in practical scenario. Though there are three points that should be kept in mind:

1. Image Registration is extremely essential to successful Superresolution.
2. If the low-res images have redundant data, then Superresolution does not work very well.
3. If the low-res images have all the information about the high-res, i.e. they have zero redundant data among them, then Superresolution works perfectly

One important fact about the algorithms used in this project is that they don't require tweaking of many parameters, so same code works well for a wide range of images.

This project restricted the motion between images to be global. In real-life we seldom have global motion and will mostly have multiple motions between frames, but this project still gives a very good intuition into the complexity of Superresolution and different ways to solve it.

6. FUTURE WORK

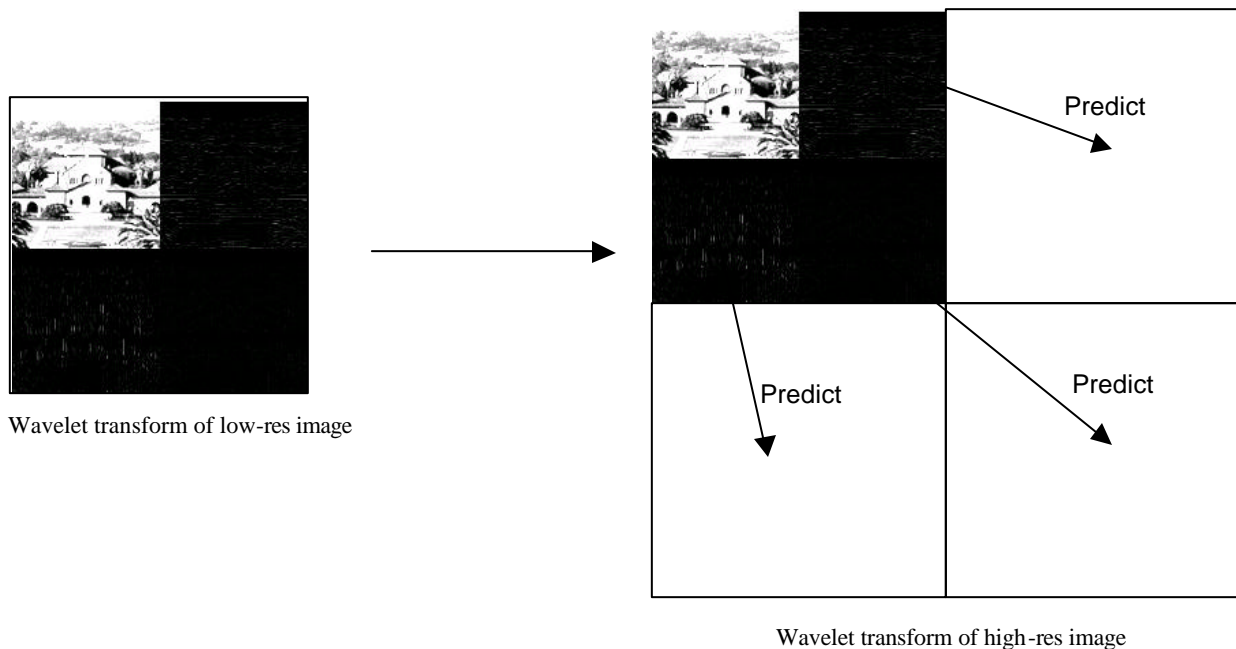
Current work from this project can be easily extended to improve performance and lessen the constraints on motion.

Possible improvements:

1. Incorporating multiple motion between frames
2. Estimating the high-frequency values more accurately in the HR image
3. Using both of the above to create a high-res video from a low-res one.

There are lots of papers on how to estimate multiple motion between frames. The good part about using Papoulis-Gerchberg algorithm is that it is independent of motion estimation technique, and so motion can be estimated separately and the second part repeated for projection onto HR grid.

High-freq values in the high-res image can be estimated using wavelet transform. Wavelet transform gives us horizontal, vertical and diagonal edge information. This information can be intelligently extended in the high-freq bins to make the edges look sharper. Though this is not a very easy task, but is still achievable. Following figure briefly describes the idea.



ACKNOWLEDGEMENTS

1. Prof. John Apostolopoulos
2. Prof. Susie Wee
3. Patrick Vandewalle

REFERENCES

1. S.M.Park, M.K.Park, M.G.Kang, *Super-Resolution Image Construction: A Technical Overview*, IEEE signal processing magazine 2003
2. P. Vandewalle, S. Süssstrunk, M. Vetterli, *Superresolution images reconstructed from aliased images*, Proc. SPIE/IS&T Visual Communications and Image Processing Conference, July 2003, Vol. 5150, p. 1398-1405, Lugano, Switzerland.
3. P. Vandewalle, S. Süssstrunk and M. Vetterli, *Double resolution from a set of aliased images*, accepted to Proc. IS&T/SPIE Electronic Imaging 2004: Sensors and Camera Systems for Scientific, Industrial, and Digital Photography Applications V, January 2004, Vol. 5301.
4. Z. Jiang, T.T. Wong and H. Bao. *Practical Super-Resolution from Dynamic Video Sequences*. In Proceedings of IEEE Computer Vision and Pattern Recognition 2003 (CVPR 2003), Vol.2. Pages 549~554, Madison, Wisconsin, USA, June 16~22, 2003.
5. P.J.S.G. Ferreira, *Teaching signal and image reconstruction algorithms*
6. A. Papoulis, *A New Algorithm in Spectral Analysis and Band-Limited Extrapolation*, IEEE Transactions on Circuits and Systems 22(9), pp. 735-742, 1975
7. R.W. Gerchberg, *Super-resolution through error energy reduction*, Optica Acta 21(9), pp. 709-720, 1974