Image Reconstruction from Non-Uniformly Sampled Spectral Data: Midterm Progress Report

Alfredo Nava-Tudela

January, 2008

Abstract

We review the progress in the study and implementation of the ideas described in Section 6.1 – *Milestones* – of [1].

1 Data set selection

We are required to select a data set on which to perform tests and validation of the algorithms developed for the project.

The test images used for this project come from two sources. The first source is the University of Southern California's Signal & Image Processing Institute image database, whose first edition was distributed in 1977. Many new images have been added since then. This database has been widely used for image processing benchmarking.

However, it can be argued that they have outlived their usefulness given their size. For this reason, a second set of images is selected from the The New Test Images database of the Image Compression Benchmark website. In particular, we have selected the 8 bit gray images subset.

For the mid-year project presentation, only the first database is used given the current speed of the code.

1. USC-SIPI Image Database, http://sipi.usc.edu/database/index.html

   From which we selected, from Vol. 3 - Miscellaneous, the following

Filename	Description	Size	Type
4.2.03	Mandrill (a.k.a Baboon)	512x512 pixels	Color (24 bits/pixel)
4.2.04	Girl (Lena, or Lenna)	512x512 pixels	Color (24 bits/pixel)
4.2.07	Peppers	512x512 pixels	Color (24 bits/pixel)
5.3.01	Man	1024x1024 pixels	Grayscale (8 bits/pixel)
5.3.02	Airport	1024x1024 pxls	Grayscale (8 bits/pixel)

   Note: The images are stored in the TIFF format.

2. Image Compression website at http://www.imagecompression.info/test_images/, from which we will select images contained in the gray8bit.zip file.

To summarize, we will draw images from the above two sources for our experiments.

2 Software development

Recall that we are trying to solve

$$\boldsymbol{A}\boldsymbol{f} = \boldsymbol{b}, \tag{1}$$

where

$$\boldsymbol{A} = \begin{pmatrix} a_0(\sigma_0,\gamma_0) & a_1(\sigma_0,\gamma_0) & \dots & a_{N^2-1}(\sigma_0,\gamma_0) \\ a_0(\sigma_1,\gamma_1) & a_1(\sigma_1,\gamma_1) & \dots & a_{N^2-1}(\sigma_1,\gamma_1) \\ \vdots & \vdots & & \vdots \\ a_0(\sigma_{M-1},\gamma_{M-1}) & a_1(\sigma_{M-1},\gamma_{M-1}) & \dots & a_{N^2-1}(\sigma_{M-1},\gamma_{M-1}) \end{pmatrix}, \tag{2}$$

$\boldsymbol{f}$ is the vector of intensities in the spatial domain to be found, and $\boldsymbol{b}$ is a vector of spectral values sampled at the corresponding (σ_i, γ_i) points in the spectral domain.

We solve this system of linear equations by solving

$$(\boldsymbol{A}^*\boldsymbol{A})\boldsymbol{f} = \boldsymbol{A}^*\boldsymbol{b}. \tag{3}$$

Observe that we can set

$$\boldsymbol{A} = \begin{pmatrix} a_0(\sigma_0,\gamma_0) & a_1(\sigma_0,\gamma_0) & \dots & a_{N^2-1}(\sigma_0,\gamma_0) \\ a_0(\sigma_1,\gamma_1) & a_1(\sigma_1,\gamma_1) & \dots & a_{N^2-1}(\sigma_1,\gamma_1) \\ \vdots & \vdots & & \vdots \\ a_0(\sigma_{M-1},\gamma_{M-1}) & a_1(\sigma_{M-1},\gamma_{M-1}) & \dots & a_{N^2-1}(\sigma_{M-1},\gamma_{M-1}) \end{pmatrix} = \begin{pmatrix} \boldsymbol{v}_0^T \\ \boldsymbol{v}_1^T \\ \vdots \\ \boldsymbol{v}_{M-1}^T \end{pmatrix}, \tag{4}$$

where the vector $\boldsymbol{v}_i^T = (a_0(\sigma_i,\gamma_i), a_1(\sigma_i,\gamma_i), \dots, a_{N^2-1}(\sigma_i,\gamma_i))$ is the i-th row of $\boldsymbol{A}$. From this, it is easy to see that

$$\begin{aligned} \boldsymbol{A}^*\boldsymbol{A} &= \begin{pmatrix} \sum_{k=0}^{M-1} \overline{a_0(\sigma_k,\gamma_k)} a_0(\sigma_k,\gamma_k) & \dots & \sum_{k=0}^{M-1} \overline{a_0(\sigma_k,\gamma_k)} a_{N^2-1}(\sigma_k,\gamma_k) \\ \vdots & & \vdots \\ \sum_{k=0}^{M-1} \overline{a_{N^2-1}(\sigma_k,\gamma_k)} a_0(\sigma_k,\gamma_k) & \dots & \sum_{k=0}^{M-1} \overline{a_{N^2-1}(\sigma_k,\gamma_k)} a_{N^2-1}(\sigma_k,\gamma_k) \end{pmatrix}, \\ &= \sum_{k=0}^{M-1} \begin{pmatrix} \overline{a_0(\sigma_k,\gamma_k)} a_0(\sigma_k,\gamma_k) & \dots & \overline{a_0(\sigma_k,\gamma_k)} a_{N^2-1}(\sigma_k,\gamma_k) \\ \vdots & & \vdots \\ \overline{a_{N^2-1}(\sigma_k,\gamma_k)} a_0(\sigma_k,\gamma_k) & \dots & \overline{a_{N^2-1}(\sigma_k,\gamma_k)} a_{N^2-1}(\sigma_k,\gamma_k) \end{pmatrix}, \\ &= \sum_{k=0}^{M-1} \begin{pmatrix} \overline{a_0(\sigma_k,\gamma_k)} \\ \overline{a_1(\sigma_k,\gamma_k)} \\ \vdots \\ \overline{a_{N^2-1}(\sigma_k,\gamma_k)} \end{pmatrix} (a_0(\sigma_k,\gamma_k), a_1(\sigma_k,\gamma_k), \dots, a_{N^2-1}(\sigma_k,\gamma_k)), \\ &= \sum_{k=0}^{M-1} \boldsymbol{v}_k \boldsymbol{v}_k^*. \end{aligned}$$

That is, $\boldsymbol{A}^*\boldsymbol{A}$ is the sum of M rank-1 updates. Also, it is easy to see that

$$\boldsymbol{A}^*\boldsymbol{b} = \begin{pmatrix} \sum_{k=0}^{M-1} \overline{a_0(\sigma_k,\gamma_k)} b_k \\ \vdots \\ \sum_{k=0}^{M-1} \overline{a_{N^2-1}(\sigma_k,\gamma_k)} b_k \end{pmatrix} = \sum_{k=0}^{M-1} b_k \boldsymbol{v}_k,$$

where $\boldsymbol{b}^T = (b_0, b_1, \dots, b_{M-1})$. Hence, our current algorithm to compute $\boldsymbol{A}^*\boldsymbol{A}$ and $\boldsymbol{A}^*\boldsymbol{b}$ is:

Initialize the variables

- $AstarA$ is the $N^2 \times N^2$ zero matrix
- V is the $1 \times N^2$ zero vector
- B is the $M \times 1$ vector with the given spectral data
- $AstarB$ is the $N^2 \times 1$ zero vector

1. Let $V = (a_0(\sigma_0, \gamma_0), \ldots, a_{N^2-1}(\sigma_0, \gamma_0))^T$ (we'll need a "for loop" of length N^2 here)
2. $AstarA = VV^*$
3. $AstarB = b_0 V$
4. For $j = 1$ to $M - 1$
   - Let $V = (a_0(\sigma_j, \gamma_j), \ldots, a_{N^2-1}(\sigma_j, \gamma_j))^T$ (need a "for loop" of length N^2 to make V)
   - $AstarA \doteq AstarA + VV^*$
   - $AstarB \doteq AstarB + b_j V$

   end

So, the algorithm requires memory for $N^2 \times N^2 + M \times 1 + N \times 1 + N \times 1$ scalars.
The computational cost requires

- M "for loops" of length N^2 to create the vectors $\boldsymbol{v}_i$
- A "for loop" of length M to create $\boldsymbol{A}^*\boldsymbol{A}$ and $\boldsymbol{A}^*\boldsymbol{b}$

Once we obtain $\boldsymbol{A}^*\boldsymbol{A}$ and $\boldsymbol{A}^*\boldsymbol{b}$, we use Matalb's \ operator to solve the least squares problem defined by equation 1. With this algorithm it is possible to establish that the theory in [1] is actually capable of reconstructing images as shown in the next section.

3 Proof of concept

To validate the theory, we create synthetic data by choosing a set of points in the spectral domain from which we sample the high resolution image to create its Fourier transform at those points. The set of points is represented by a *geometry file* and the image is needed to compute exactly, since we have an analytic formula for such purpose, its Fourier transform values. See Figure 1.

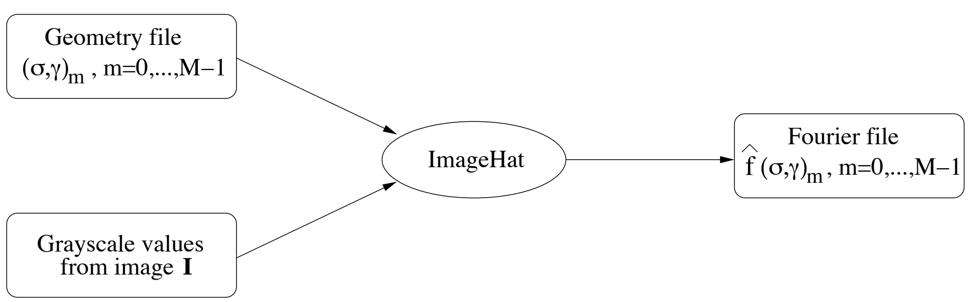

Figure 1: Creation of synthetic data from database image.

Once we have the synthetic data, we can use the algorithm described in the previous section to generate $\boldsymbol{A}^*\boldsymbol{A}$ and $\boldsymbol{A}^*\boldsymbol{b}$. We need, once again, the geometry file, and we use the Fourier file generated in the previous step. See Figure 2.

Finally, we can use $\boldsymbol{A}^*\boldsymbol{A}$ and $\boldsymbol{A}^*\boldsymbol{b}$ and any adequate solver to obtain a reconstruction of the initial image at a lower resolution, as determined by the number of points in the spectral domain used to sample the Fourier transform of the original high resolution image. See Figure 3.

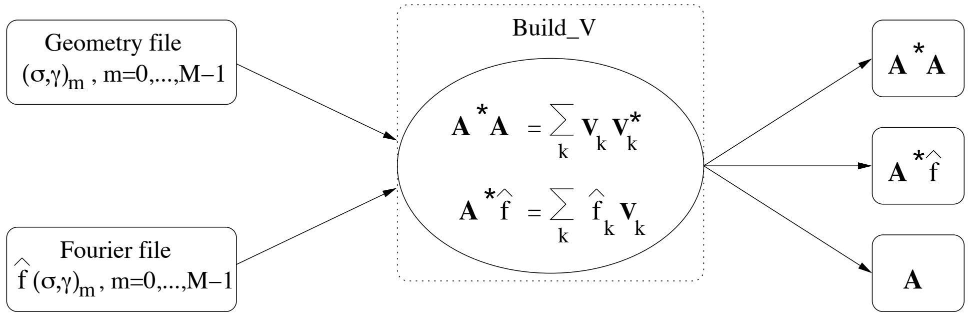

Figure 2: Construct $\boldsymbol{A}^*\boldsymbol{A}$ and $\boldsymbol{A}^*\boldsymbol{b}$.

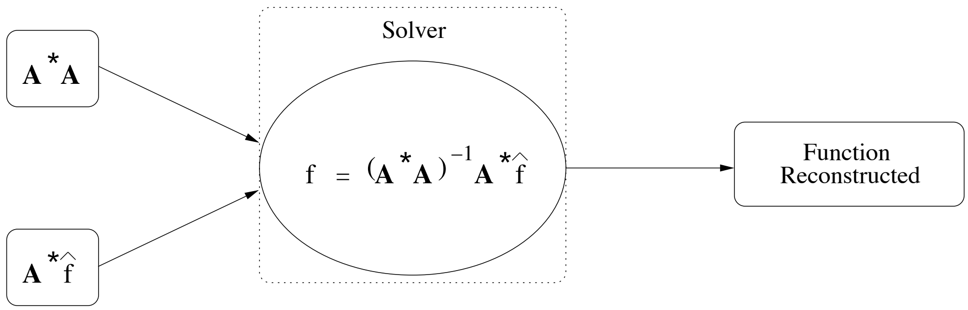

Figure 3: Reconstruct image using $\boldsymbol{A}^*\boldsymbol{A}$ and $\boldsymbol{A}^*\boldsymbol{b}$.

4 Validation

To validate the algorithm, we carried out the steps described above for a subset of the images from our data set, and we down sampled in the spatial domain the original high resolution images –by doing simple 4-neighbor averaging–, and compared this with the results of the reconstruction. One can see the clear resemblance of both procedures, however this does not explain the differences found. This needs to be explored.

References

[1] Alfredo Nava-Tudela. Image reconstruction from non-uniformly sampled spectral data. AMSC 663 Class Project Proposal, 2008.

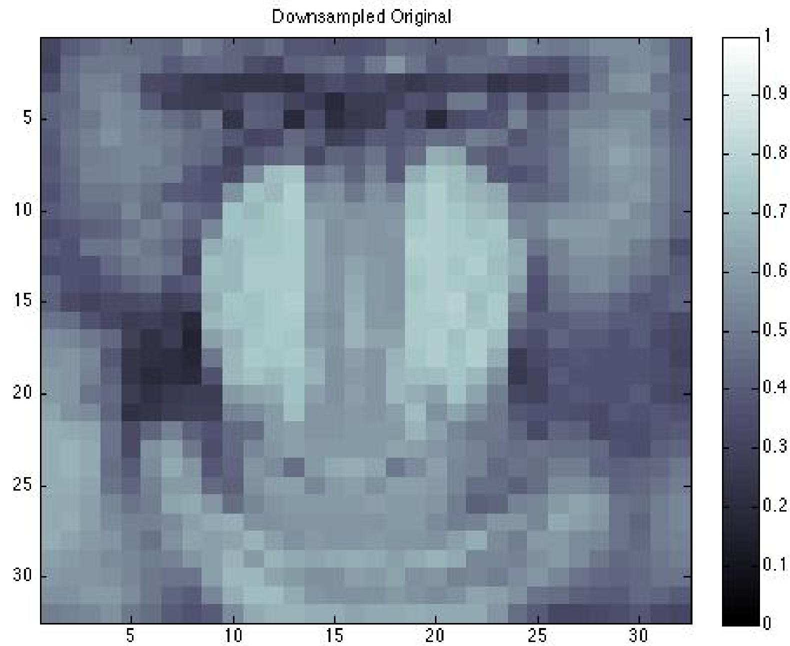

Figure 4: Downsampled baboon to 32x32 pixels.

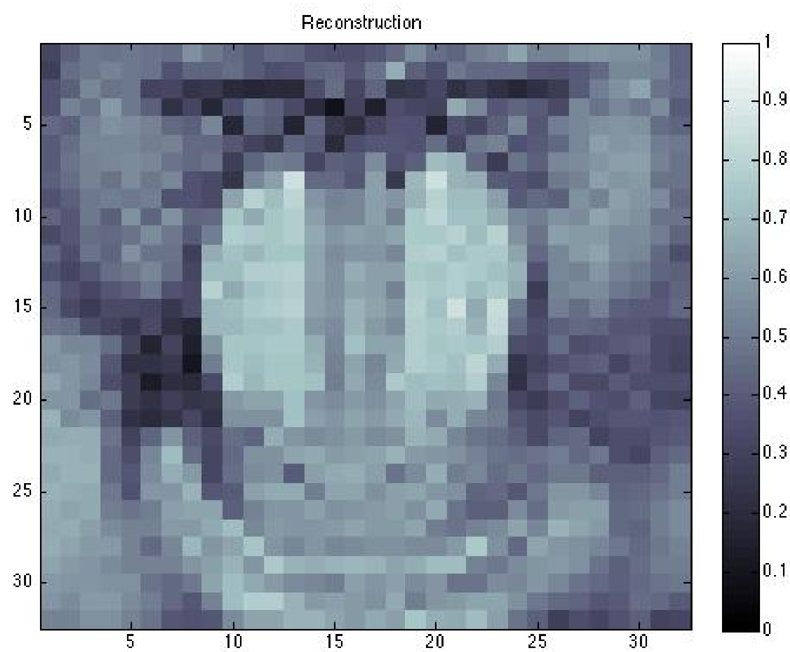

Figure 5: Reconstruction of baboon at same resolution by our algorithm.

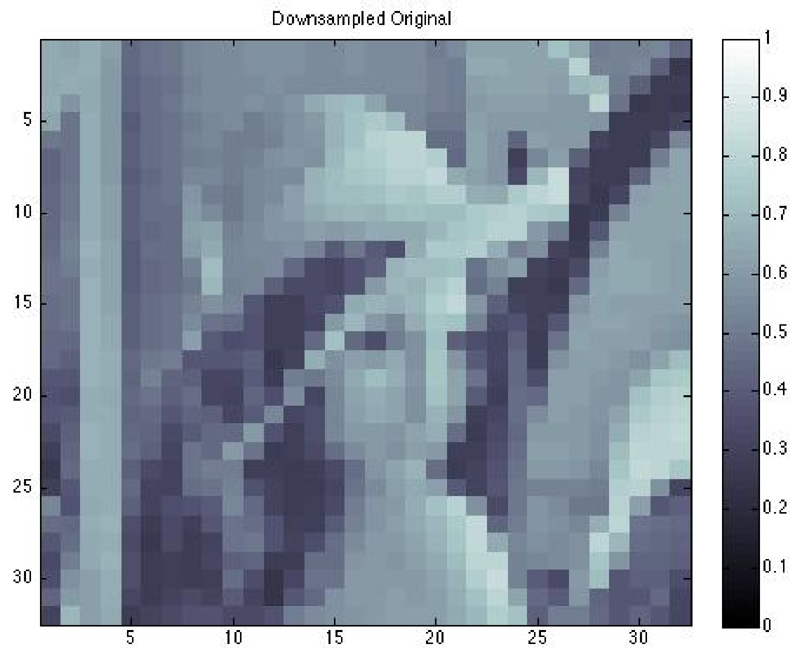

Figure 6: Downsampled Lena to 32x32 pixels.

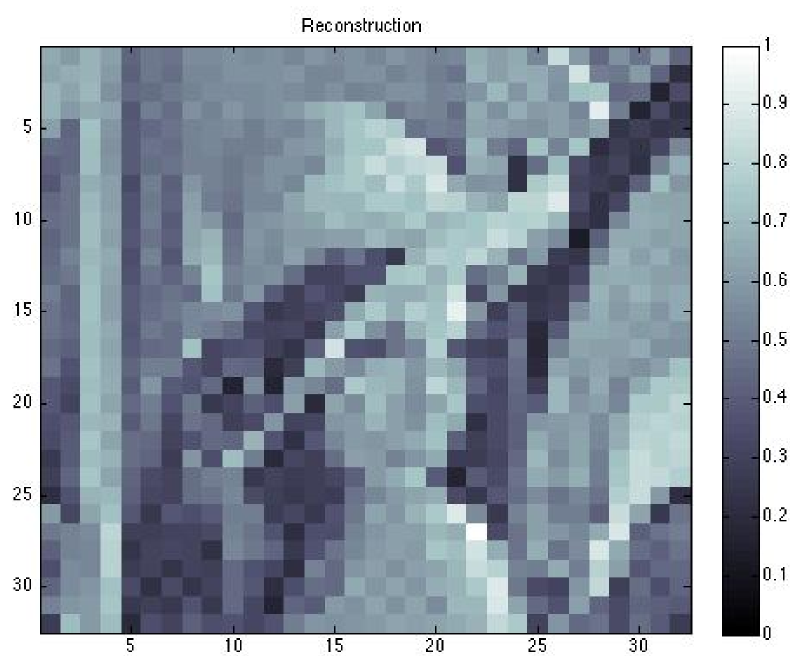

Figure 7: Reconstruction of Lena at same resolution by our algorithm.

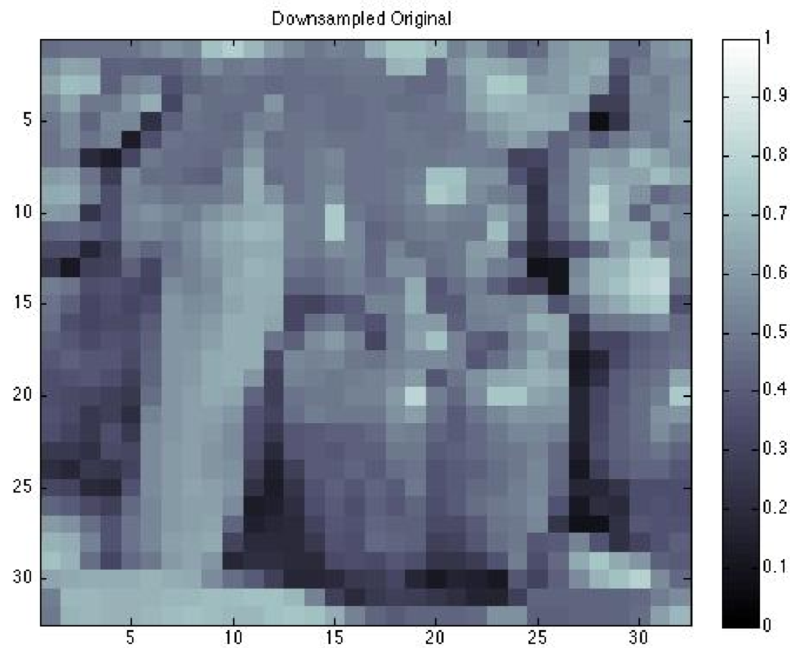

Figure 8: Downsampled peppers to 32x32 pixels.

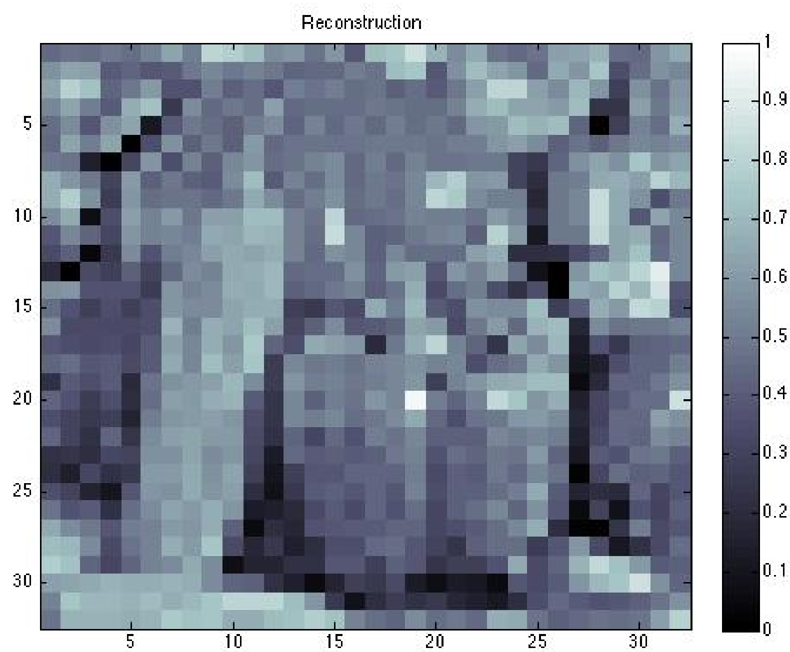

Figure 9: Reconstruction of peppers at same resolution by our algorithm.