**Replication of Results of the Dadkhahi, Gotchev, and Egiazarian Inverse Polynomial Reconstruction Method (IPRM) in the DCT Domain**

Text

Description automatically generated with medium confidence

Alex Reigle

CEG 7850 Digital Image Processing

Wright State University

Instructor: Dr. Arnab Shaw

December 6, 2021

**Table of Contents**

Declaration Statement …………………………………………………………………………………………………………... 3

Abstract …………………………………………………………………………………………………………………………………. 4

Introduction …………………………………………………………………………………………………………………………… 5

Technical Discussion ………………………………………………………………………………………………………………. 6

Results …………………………………………………………………………………………………………………………………… 7

Discussion of Results …………………………………………………………………………………………………………..… 24

Discussion ………………..…………………………………………………………………………………………………………… 25

Figures …………………………………………………………………………………………………………………………………… 26

References …………………………………………………………………………………………………………………………….. 27

Appendix ……………………………………………………………………………………………………………………………….. 28

**Declaration Statement**

I hereby declare that this Report and the Matlab codes were written/prepared entirely by me based on my own work, and I have not used any material from another Project at another department/ university/college anywhere else, including Wright State. I also declare that I did not seek or receive assistance from any other person and I did not help any other person to prepare their reports or code.  The report mentions explicitly all sources of information in the reference list. I am aware of the fact that violation of these clauses is regarded as cheating and can result in invalidation of the paper with zero grade. Cheating or attempted cheating or assistance in cheating is reportable to the appropriate authority and may result in the expulsion of the student, in accordance with the University and College Policies.

**1 Abstract**

A partial replication of results of the “Inverse Polynomial Reconstruction Method in [the] DCT Domain” has been completed and is shown below. A method to denoise a signal compressed via truncation has been created to emulate the IPRM technical detailed in the primarily referenced article. Both 1D signals used to demonstrate the efficacy of the IPRM technique have been recreated (Fig. 1 and Fig. 2) and correspond to Figure 2(a-b) and Figure 4(a-b) from the article[1].

An IPRM technique has been recreated to recover a function from a special kind of corrupted signal, that is a signal corrupted by the Gibbs Phenomenon. This technique is applicable to both 1D signals and images and is demonstrated on both the signals from the referenced article and images that are built-in for MatLab R2020b and MatLab R2021b. Gibbs phenomenon effects and the removal thereof are discussed.

Additional discussion is included to elaborate on progress made and short any short comings of the developed IPRM algorithm. Furthermore, suggestions for next steps are included.

**2 Introduction**

A primary motivation for image compression is to facilitate the transmission of images and reduce the energy required to do so. In many situations, whether due to a processing technique or hardware shortcomings, compression can result in a periodic noise introduction known as the Gibbs Phenomenon. The Gibbs Phenomenon is especially troublesome in situations involving orthogonal series representations (for example, Fourier or Cosine transforms) of the original image (or signal).[2] These noise artifacts are even more pronounced in images or signals that are aperiodic and have sharp discontinuities[1]. In this paper, the Gibbs Phenomenon is considered as a result of the frequency domain truncation of the discrete cosine transform (DCT).

The resolution (or removal) of the Gibbs Phenomenon is the focus of this project and much research[1]-[6]. This resolution in DCT processing systems is of particular interest[1] due to the common use of DCT transformations in image and signal compression. To accomplish this, the compressed image or signal must have a *suitable* change in basis upon the inverse transform. A polynomial basis often considered in this transformation set is that of the Gegenbauer Polynomial[6]. Where there are two solution possibilities associated with Gegenbauer polynomials.

The first of which is called the direct method (or the Gegenbauer reconstruction method) and the second is called the inverse polynomial reconstruction method (IPRM). In the direct method, the frequency domain signal is reprojected onto the Gegenbauer polynomial basis set prior to transmission. Conversely, the IPRM projects the original signal onto the Gegenbauer basis set and then into the frequency domain. After the signal is encoded using either the IPRM or direct method the signal is resolved based on a known (a priori) selection of polynomial order and compression ratio, which can be thought of as an encryption key.

This project recreates the same piecewise smooth signal under consideration by [1] and, in place of the image used, assesses a subset of images that are built-in to the MatLab environment. The induced Gibbs phenomenon is shown for each considered image or signal. The IPRM process is then undergone and each considered input is then shown to have the Gibbs phenomenon noise removed. Some images show a propensity for hiding the effects of the Gibbs phenomenon, so additional noise is injected into the system to show the denoising capabilities the IPRM offers. This highlights the finding that the IPRM can serve as a noise removing (and/or noise resistant) transformation method not only when considering compressed images, but any images that undergo transmission, reception, and processing.

**3 Technical Discussion**

 The IPRM, as discussed here and in [1], hinges on the changing the basis set of the initial image to a basis set that is less impacted by noise, that of the Gegenbauer polynomial. The Gegenbauer polynomial is defined as[4]

(1)

where k is the order of the polynomial, **ξ is the instance in time (or the associated pixel value as will be shown), and λ is a value that corresponds to the dimensionality of the input function and will be addressed later.[9] The Gegenbauer Polynomial for order k=1 is defined as 2\*λ\*ξ, and for k=0 is defined equal to 1.**

A picture containing text, watch

Description automatically generated **The series expansion of the Gegenbauer Polynomial is then expressed as[1]**

(2)

where here l is the order (previously shown as k in 1).

A picture containing text

Description automatically generated The definition of the DCT expansion is given as

(3)

A picture containing text, clock, gauge

Description automatically generatedwhere *f*k is the input function, N is the number of elements in the image or signal, and β[k] is the DCT normalization factor, defined as follows.

(4)

It is noted in [1] that eqn. 3 is the function that introduces the Gibbs phenomenon into the solution image.

Text, letter

Description automatically generated From here we substitute eqn. 2 into eqn. 3 to come to the noise resistant, compressed transmission function (*f*k) of

(5)

where Nd is the number of elements in the image Xi(xn). Here I make the assertion that for all signals and images considered, the element spacing is linear and equivalent. This allows me to take advantage of the linear mapping

 (6)

resulting in no change to the inputs. This means that, in eqn. 5, Xi(xn) = **ξ (where ξ is the original input).**

**From here, the article redefines the DCT portion of eqn. 5 as W and the Gegenbauer coefficients as G, such that**

f = W.G (7)

Therefore, by taking the pseudo-inversion of W, Wt, we finally arrive at

G = Wt.f (8)

where eqn. 8 is the definition of IPRM for the DCT.

The definition of **λ is related to the dimensionality of the function being considered. There number of dimensions, Z, is defined as**

**2λ + 2 = Z (10)**

such that a 1D signal results in **λ = 1.5.**

**4 Results**

I began by considering the simplest application of the IPRM, that of a 1D signal. To validate that the results match that of the article, I recreated the original signal

 (9)

where the signal f(x) is referred to in the paper (and from here forward) as *f*1. The recreated signal *f*1, as well as the resultant frequency domain DCT of *f*1, is shown in Fig. 1. Similarly, I recreated the Heavisine function that was used for *f*2, as it is named in the article, and showed it with the frequency domain of its DCT, Fig. 2 shows the results.

From this point I introduce the Gibbs phenomenon into the signals *f*1 and *f*2. This is done by truncating the second half of the frequency domain. The resultant image may achieve a modest compressed ratio of 2. This is done for all signals and images to maintain consistency of results. The resultant truncated signals *f*1 and *f*2 are shown with the accompanying inverse DCT signals in Fig. 3 and Fig. 4, respectively.

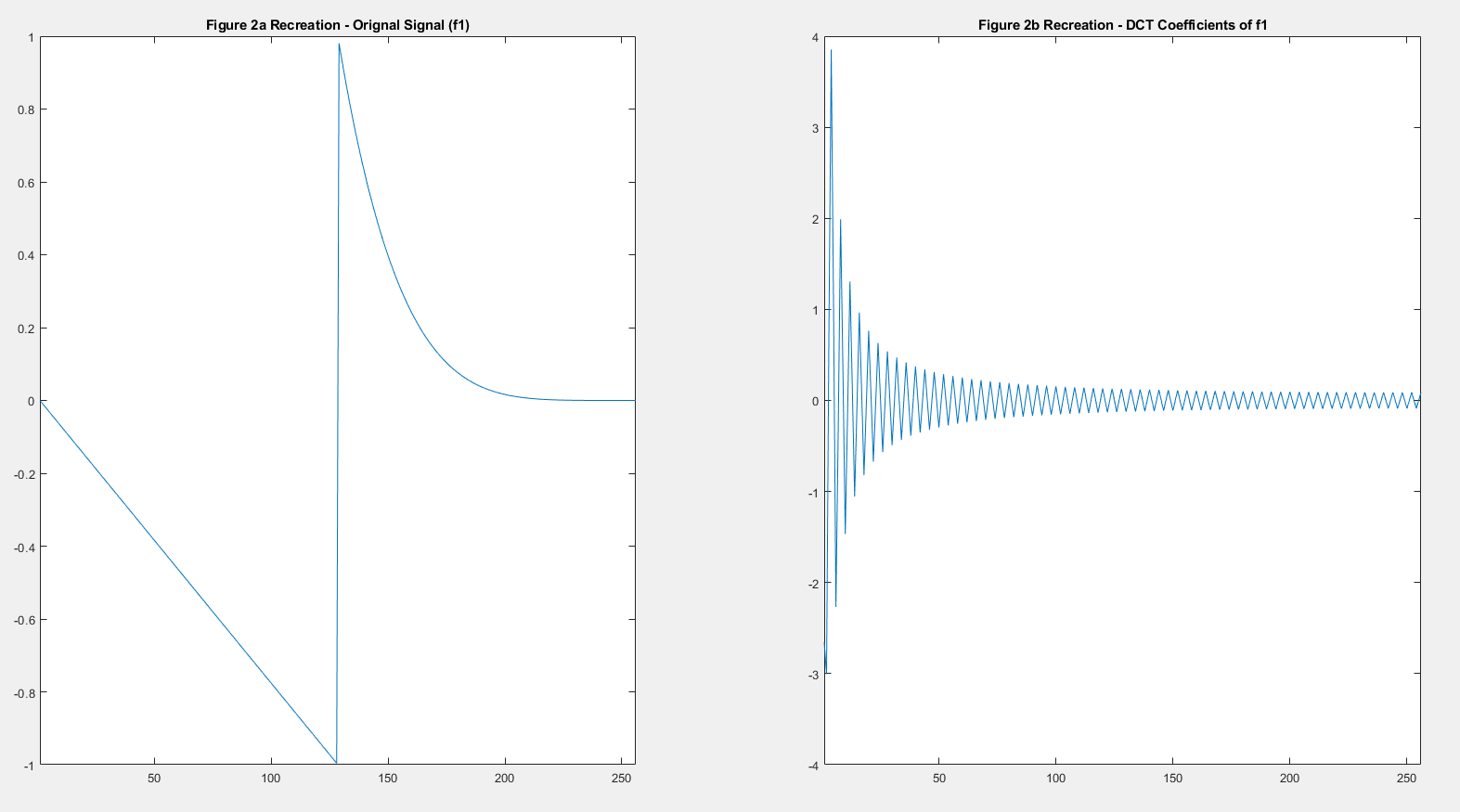


Figure 1: A recreation of Figure 2a (left) and 2b (right) from the arictle [1]. This shows the function f1 and the frequency domain of if DCT.

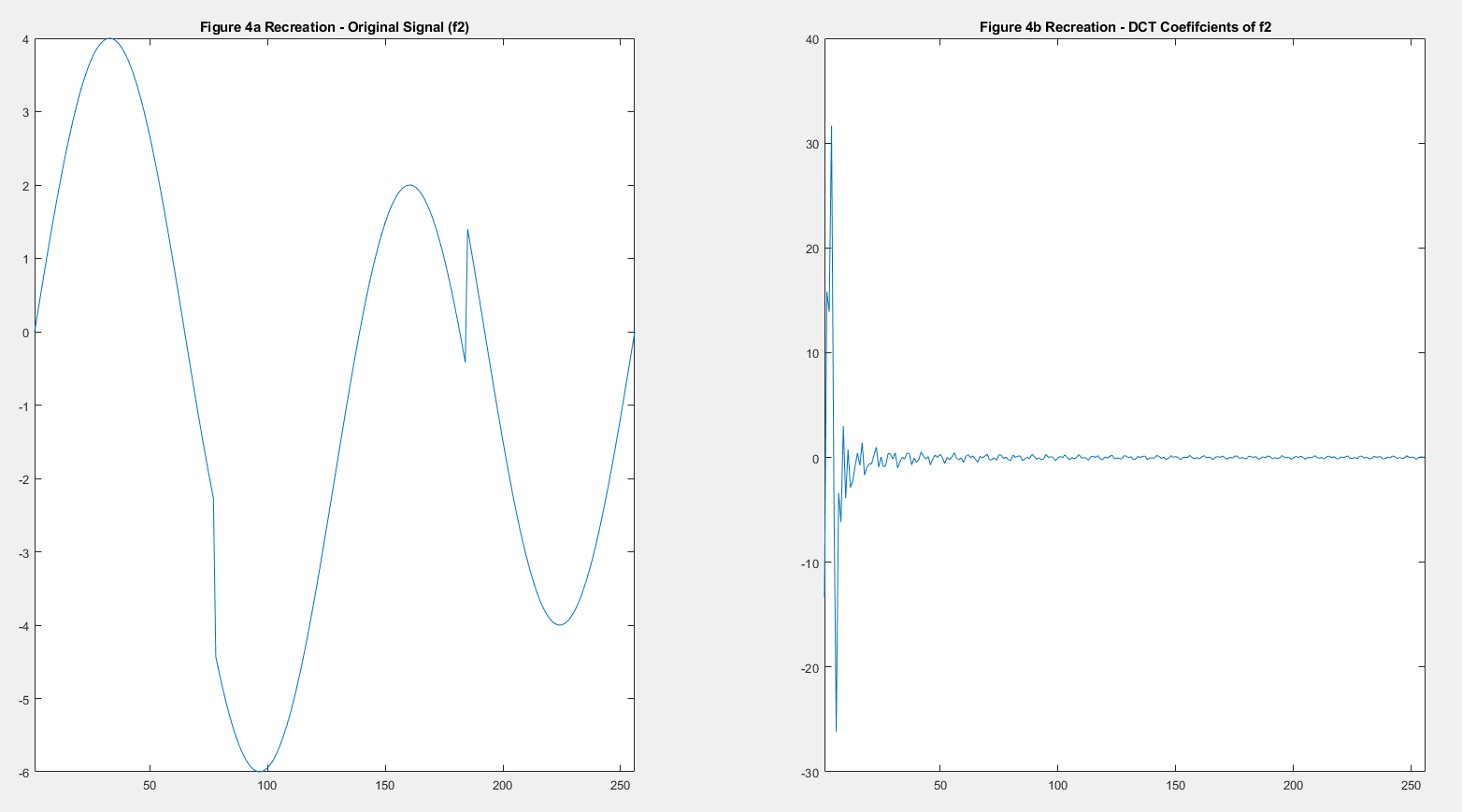


Figure 2: A recreation of Figure 4a (left) and 4b (right) from the arictle [1]. This shows the function f2 and the frequency domain of if DCT.

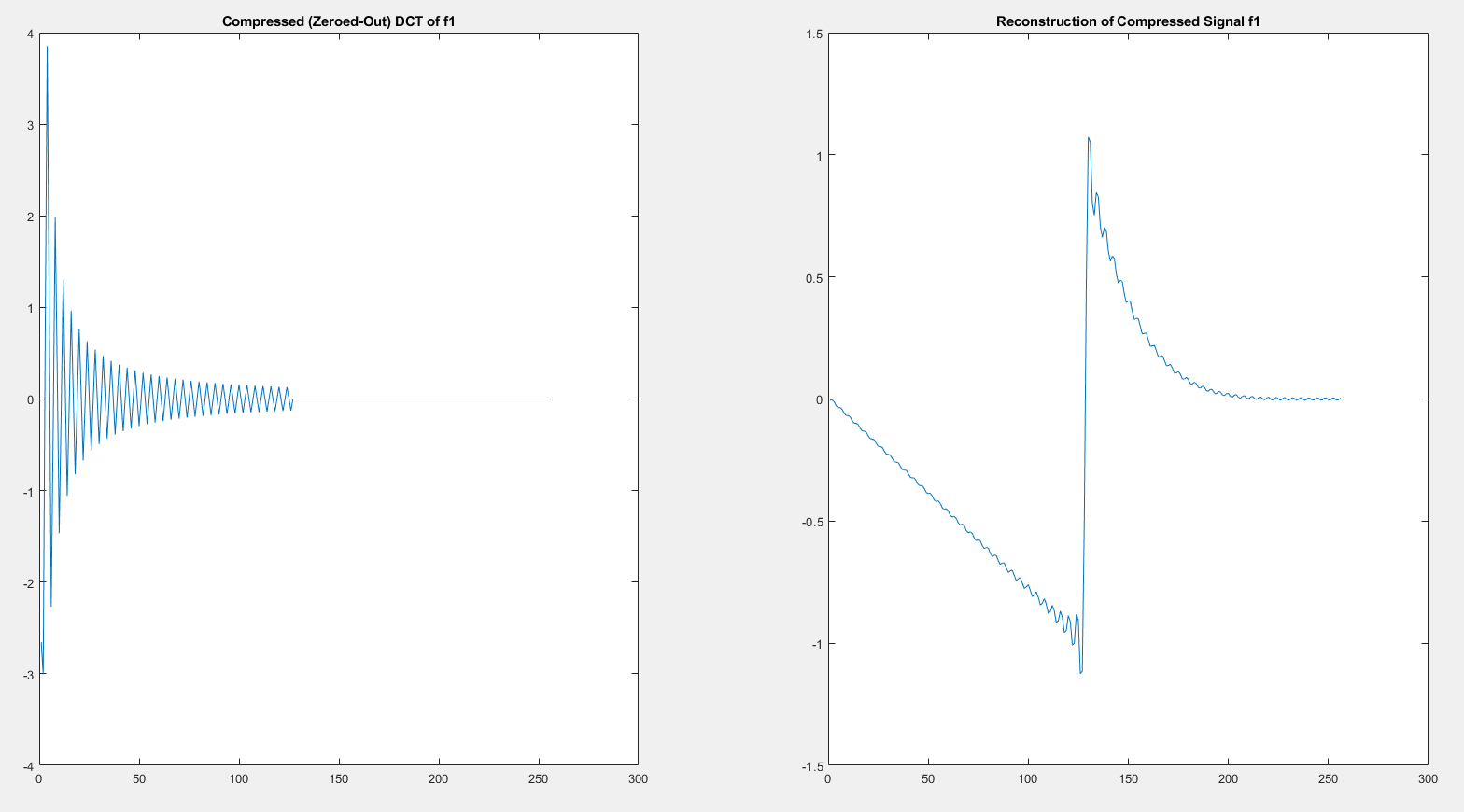


Figure 3: (a) shows the truncated signal f1 in the frequency domain and (b) shows the resultant time domain signal with the induced Gibbs phomenon.

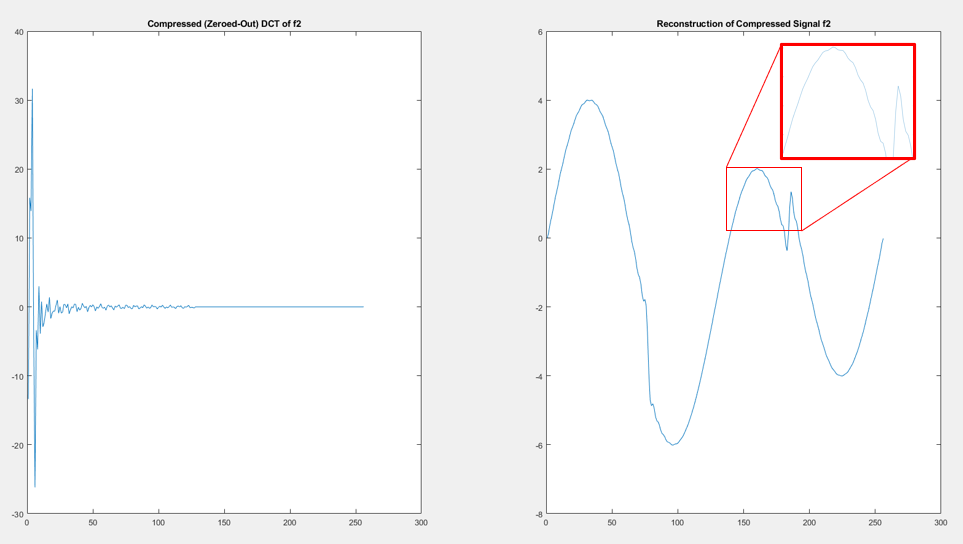


Figure 4: (a) shows the truncated signal f2 in the frequency domain and (b) shows the resultant time domain signal with the induced Gibbs phomenon.

Following the validation of inputs, I validate the output of my Gegenbauer polynomial function. I generate a linear signal spanning [-1, 1] as input and plotted the output for the first five degrees of the Gegenbauer polynomial expansion. In Fig. 5, the outputs can be seen as matching the expected results[10].

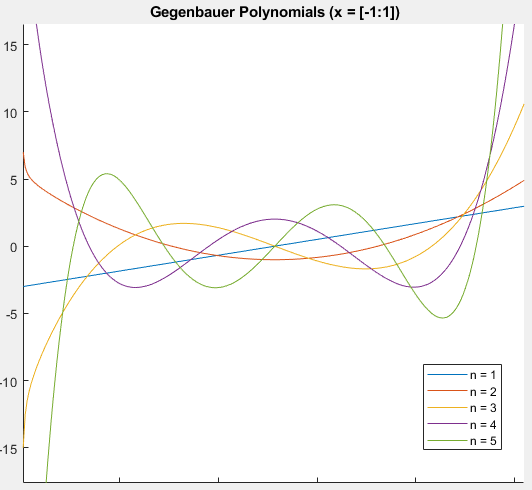


Figure 5: A plot of an N-degree Gegenbauer Polynomial for n = 1:5, that corresponds to the expected result[10].

I then validate the entire IPRM process by using the already validated signals *f*1 and *f*2. These results are shown in Fig. 6 and Fig. 7, respectively. In an analysis of the outputs, I create a root-mean-square error graph of the IPRM outputs and input. This analysis, shown in Fig. 8 and Fig. 9 (respectively), shows that the signals are both different and free of the Gibbs phenomenon.

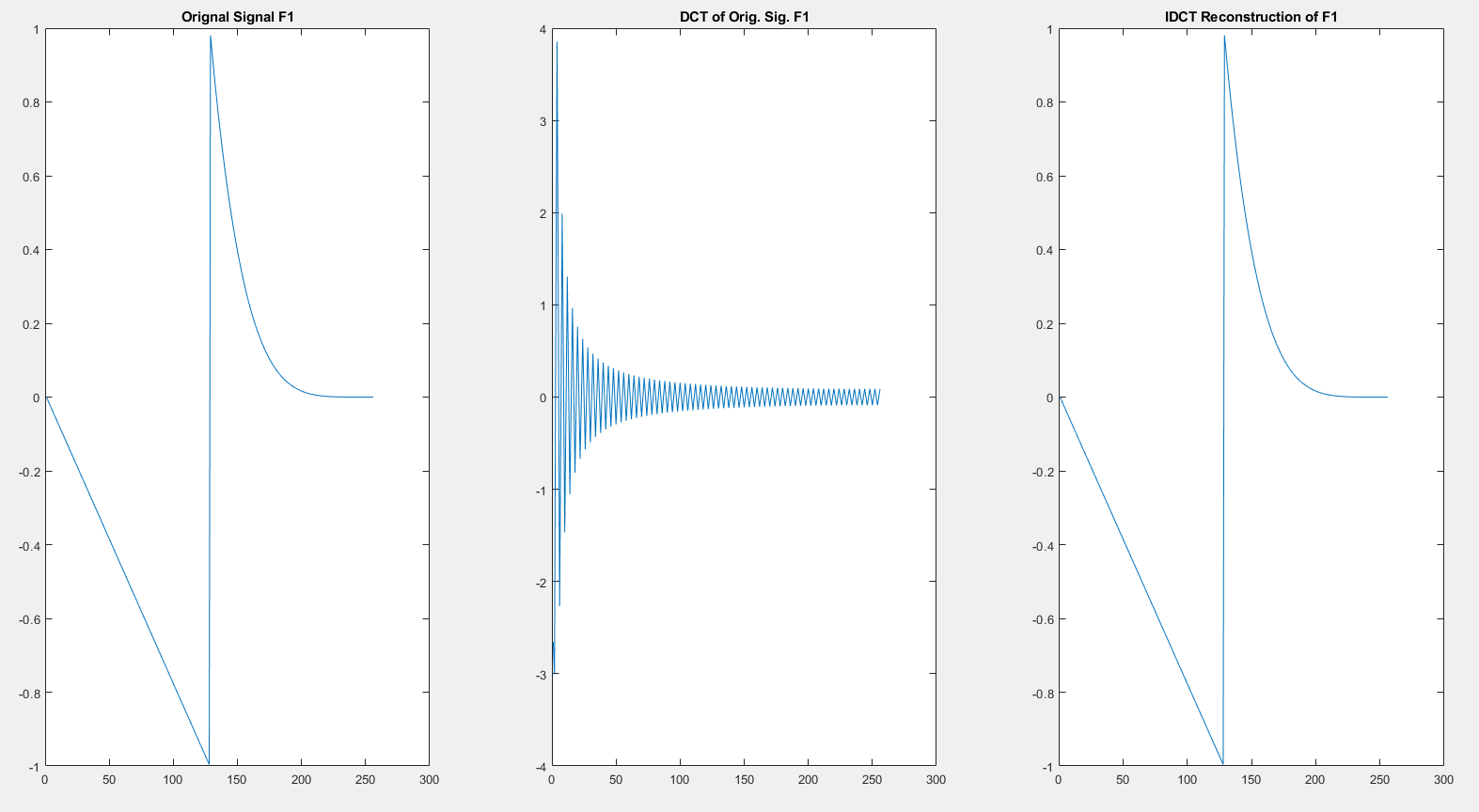


Figure 6: (a) The input signal, f1, to the IPRM. (b) The frequency domain of the input signal, f1, after truncation and change of basis to the Gegenbauer Polynomial basis set. (c) The resultant of the IPRM for input signal (b).

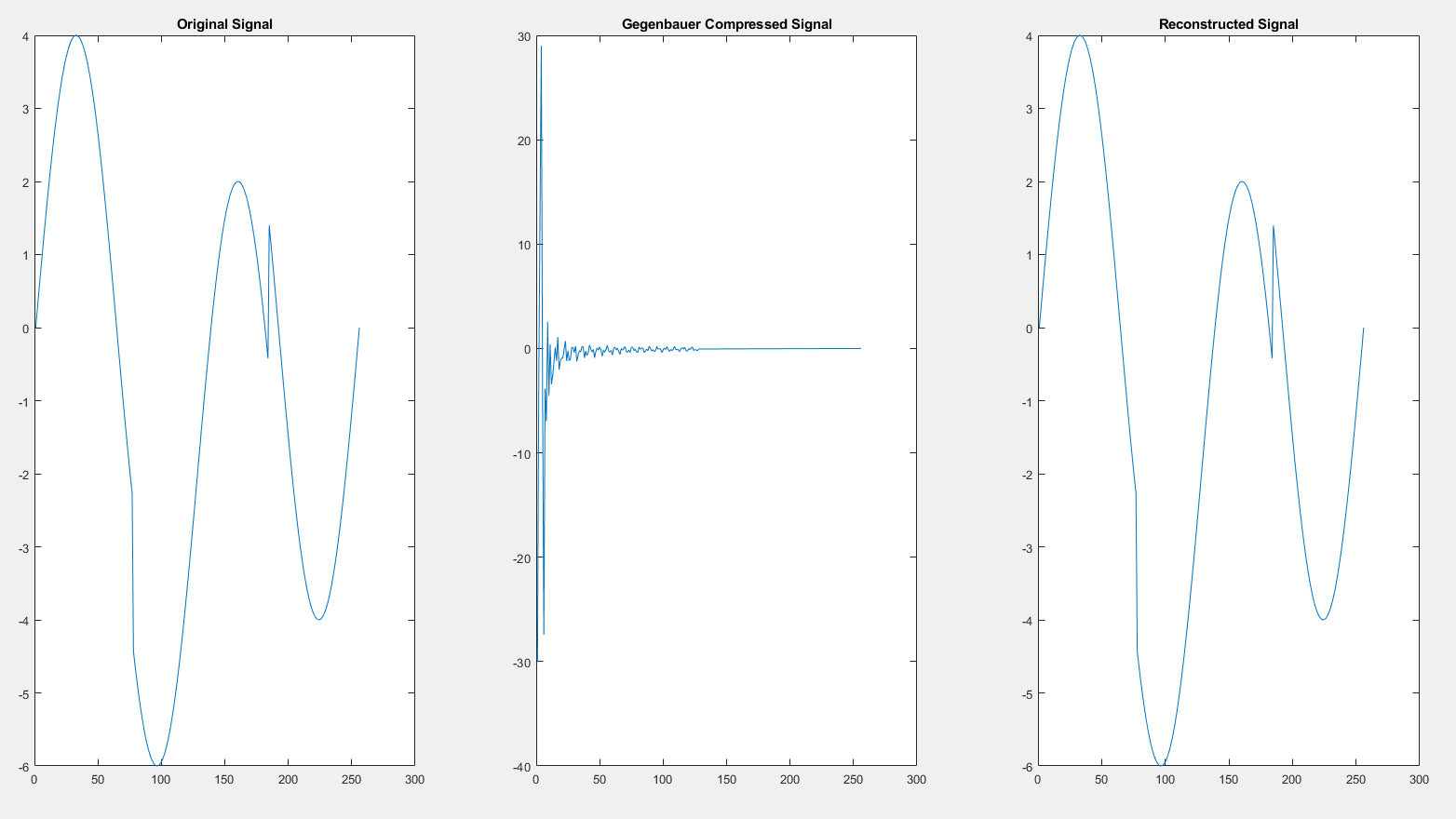


Figure 7: (a) The input signal, f2, to the IPRM. (b) The frequency domain of the input signal, f2, after truncation and change of basis to the Gegenbauer Polynomial basis set. (c) The resultant of the IPRM for input signal (b).

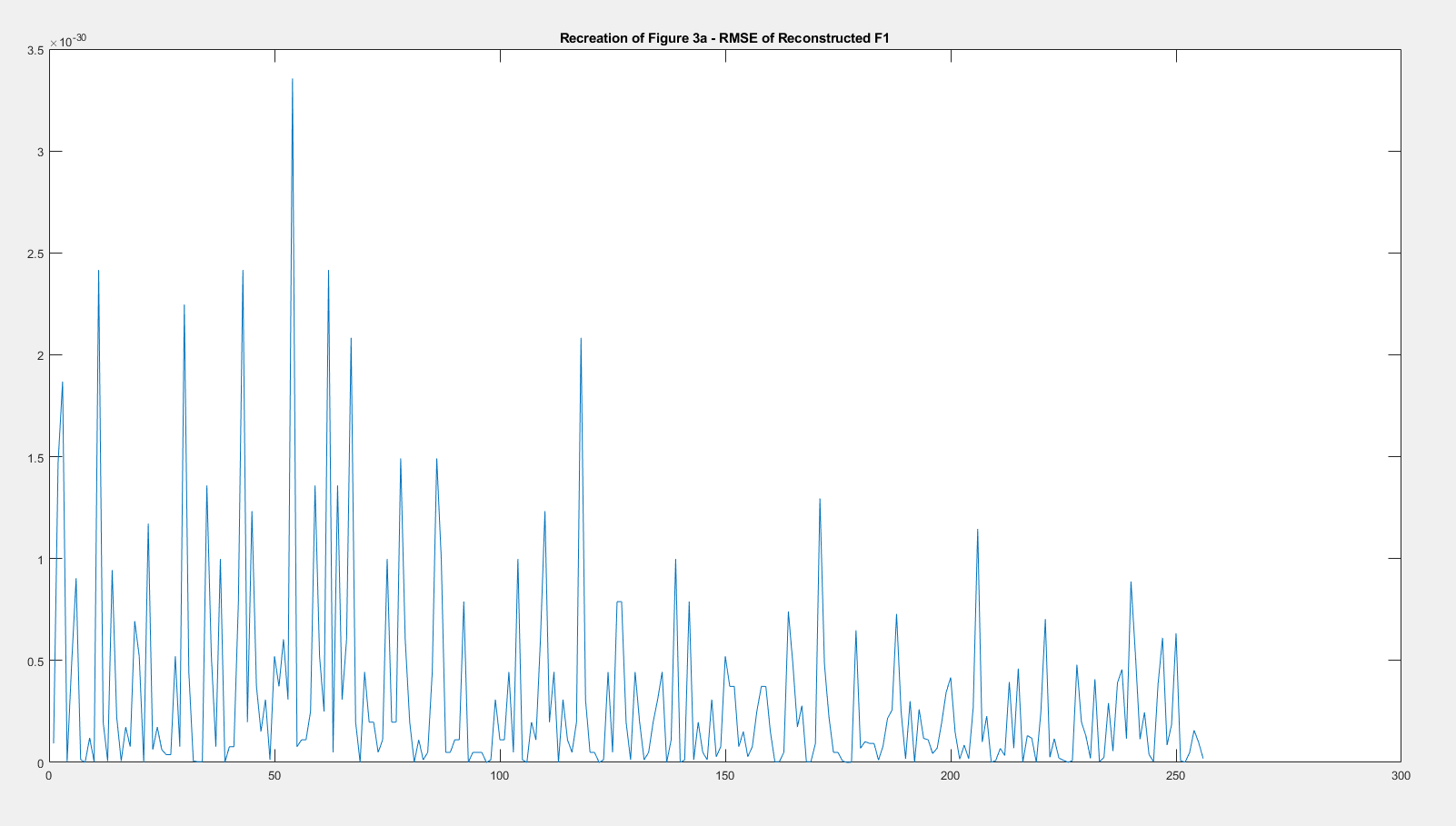


Figure 8: A recreation of Figure 3a from the reference article [1]. The root-mean-square error of the signal f1 and its corresponding IPRM output.

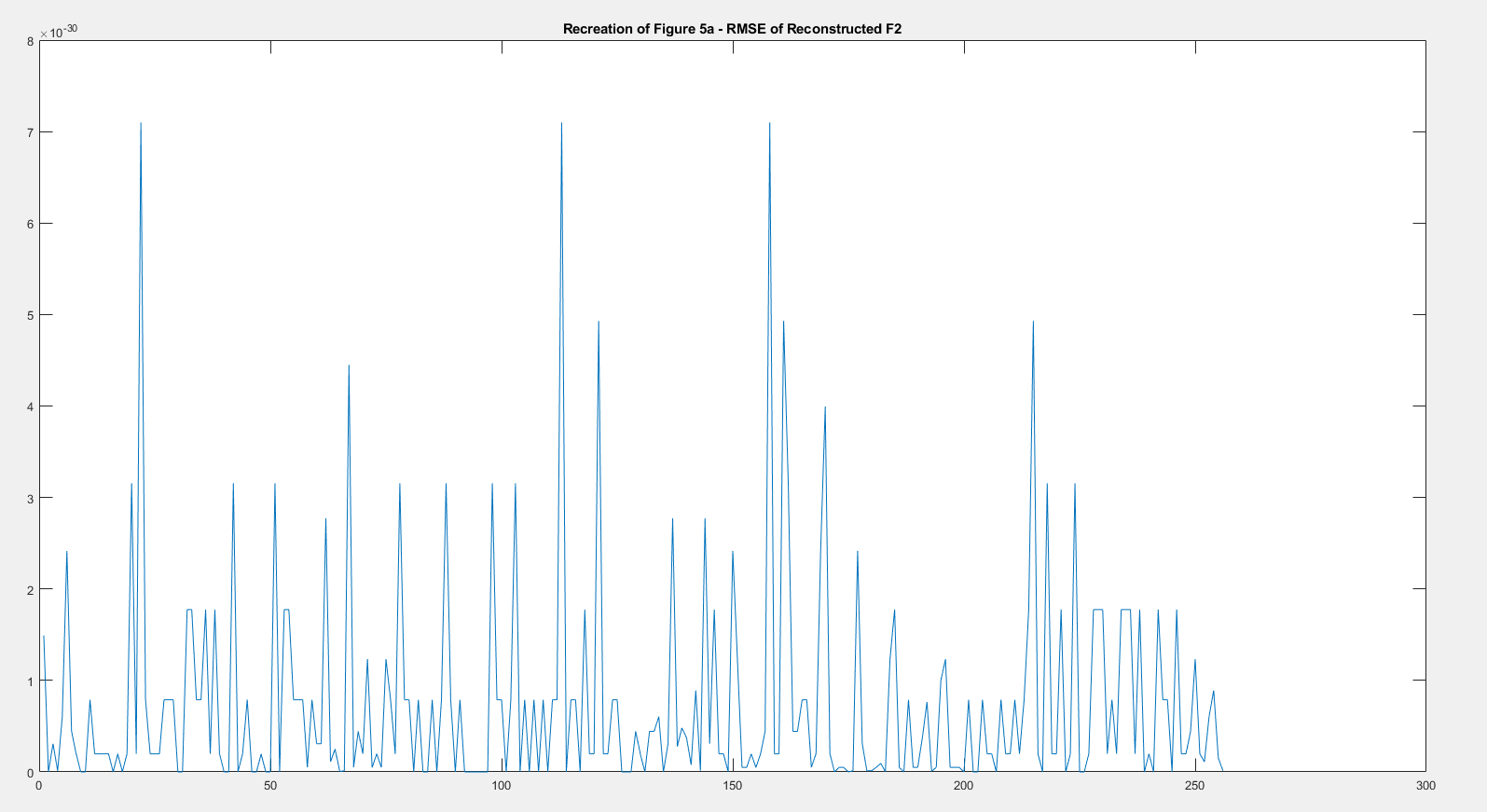


Figure 9: A recreation of Figure 5a from the reference article [1]. The root-mean-square error of the signal f2 and its corresponding IPRM output.

After validating the simplest 1D case I begin to study IPRM efficacy on 2D images, as in the progression of the article [1]. For each image considered, I plot the original image with the Gegenbauer basis set image (post compression) in the frequency domain. I also show a 1D plot showing that the image pre- and post- truncation (compression). I then show the results of the IPRM in comparison to the original image and the image corrupted by the Gibbs phenomenon. Additionally, I show the effects of the IPRM to reduce noise to a great extent at SNR levels of 3dB, 10dB, and -10dB. The noise generated is Gaussian white noise. Finally, I display the root-mean-square error of the IPRM with reference to the images input with no additional noise and the image with SNR or -10dB.

The first image considered is the ‘cameraman.tif’ image which is built into MatLab by default. In Fig. 10, I show the cameraman image, its Gegenbauer compressed image of a new basis, and a 1D representation of each in frequency domain to show the image truncation.

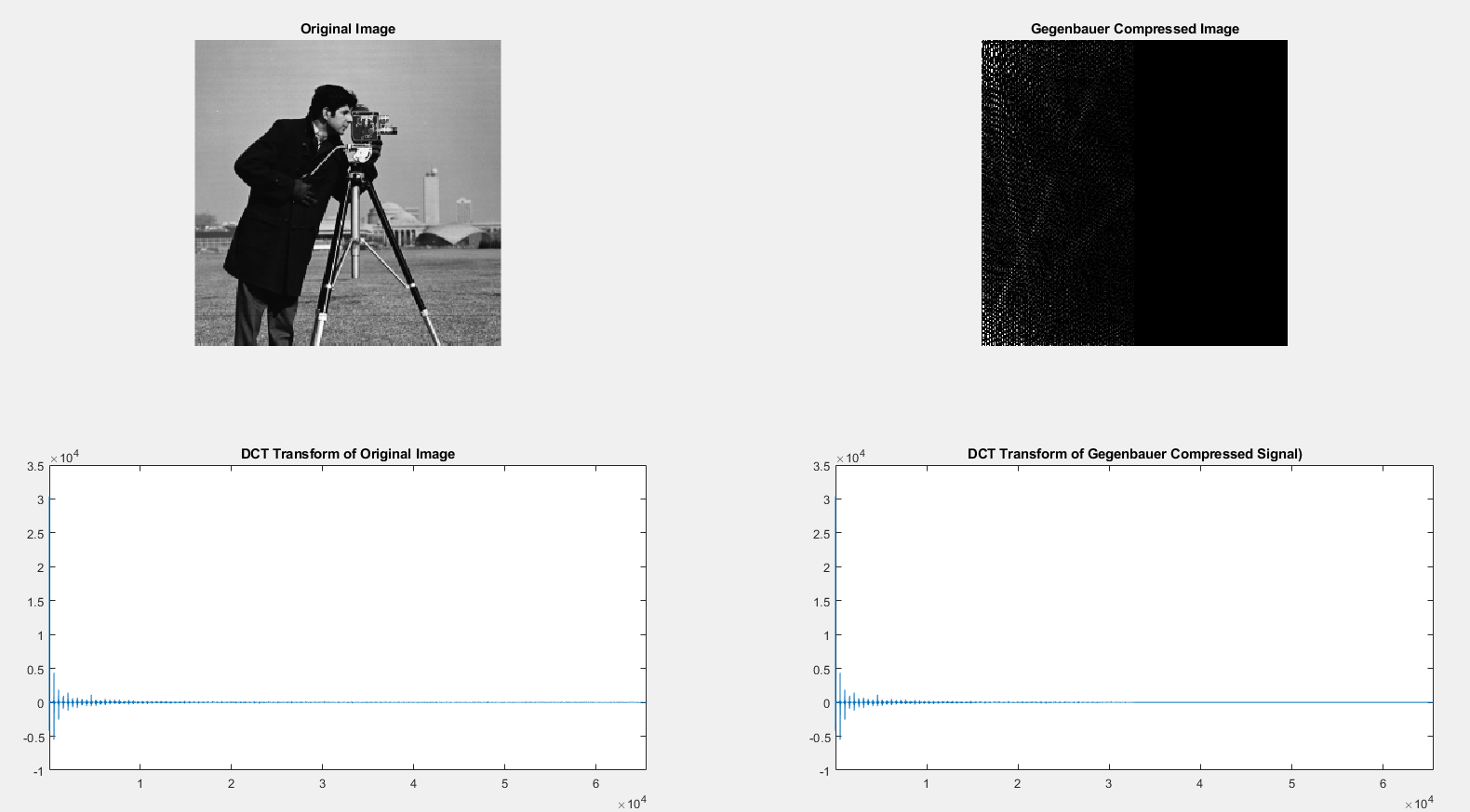


Figure 10: (a) The MatLab built-in image ‘cameraman.tif’. (b) The image (a) after Gegenbauer encoding, DCT, and truncation. (c) The 1D frequency domain of (a). (d) The 1D frequency domain of (b).

In Fig. 11, I show the cameraman image and its comparison the same image with the Gibbs phenomenon present and the results of IPRM on Figure 11b.

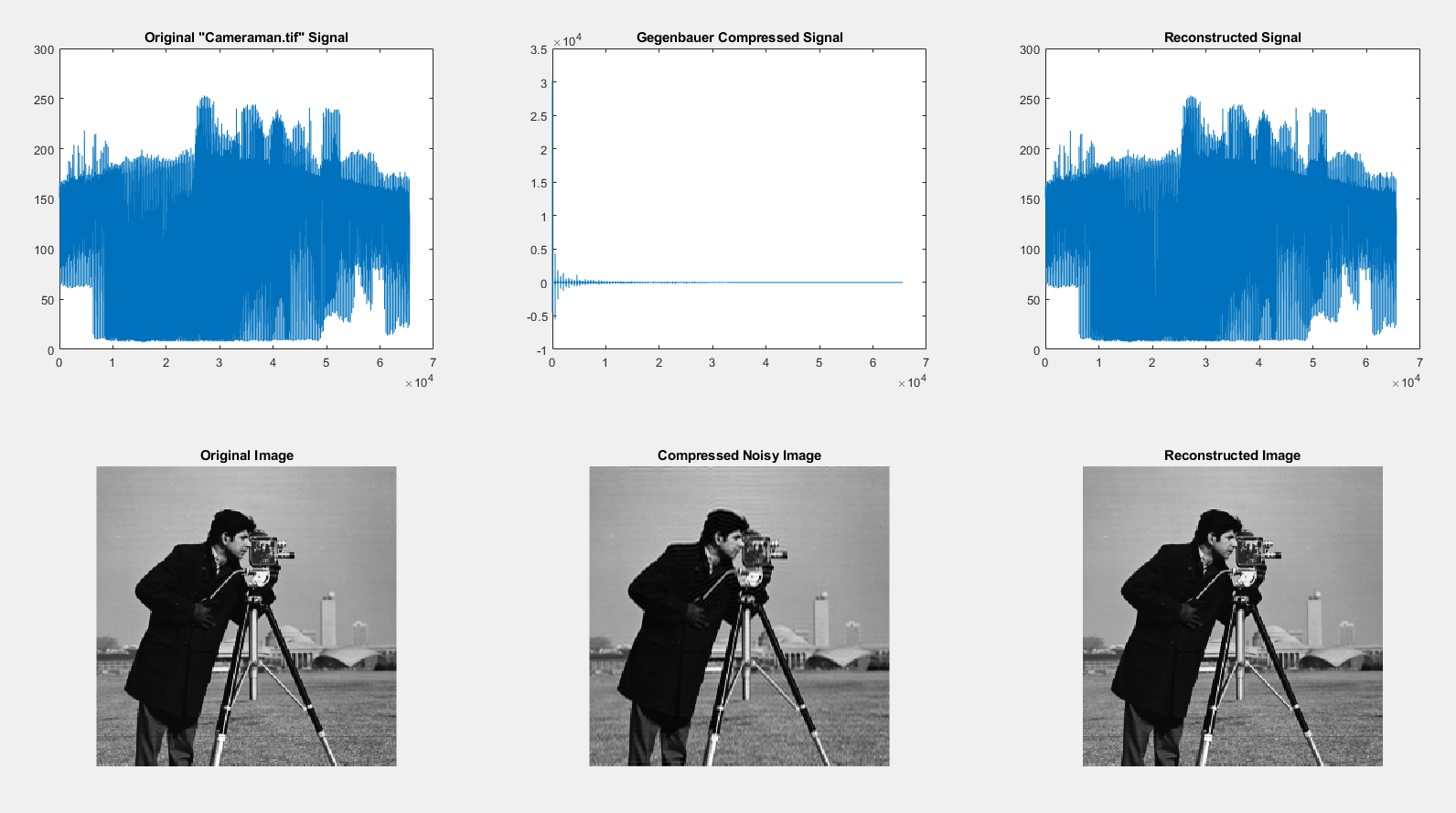


Figure 11: (a) The MatLab built-in image ‘cameraman.tif’. (b) The result of the image truncation shown in 10d. (c) The image (a) after undergoing the IPRM.

I show that the IPRM reduces noise, in Fig. 12, with images at noise levels of SNR = 3dB, 10dB. and -10dB. Then I show, in Fig. 13, the root-mean-square errors of the image reconstructed without additional noise (Fig. 11c) and the image reconstructed with an SNR = -10dB (Fig. 12f).



Figure 12: (a) Fig. 10a reconstructed without IPRM at an SNR = 3dB (b) Fig. 10a reconstructed without IPRM at an SNR = 10dB (c) Fig. 10a reconstructed without IPRM at an SNR = -10dB (d) Fig. 10a reconstructed with IPRM at an SNR = 3dB (e) Fig. 10a reconstructed with IPRM at an SNR = 10dB (f) Fig. 10a reconstructed with IPRM at an SNR = -10dB

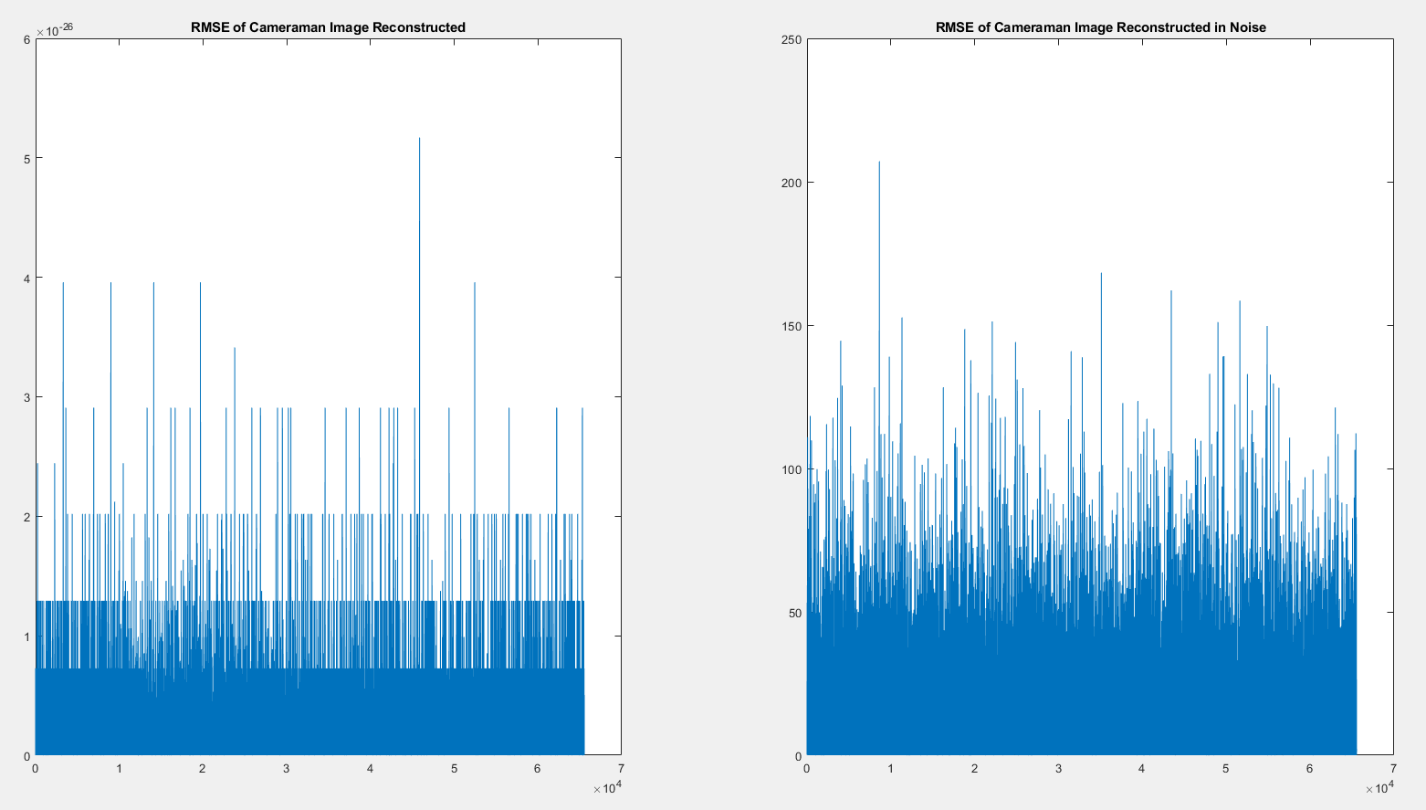
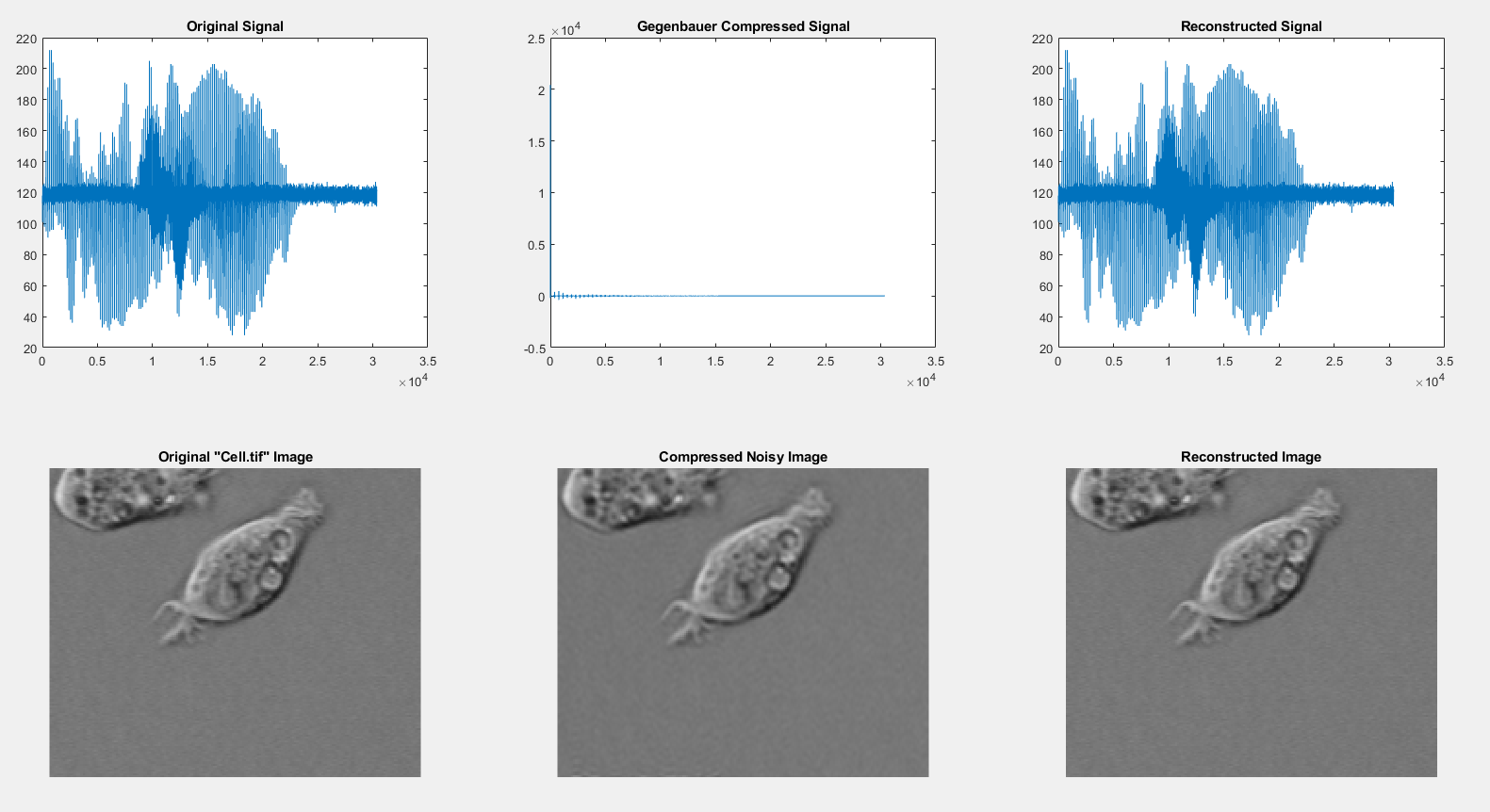


Figure 13: (a) Root-mean-square error of Fig. 10a and Fig 11c. (b) Root-mean-square error of Fig. 10a and Fig 12f.

The second image considered is the ‘cell.tif’ image which is built into MatLab by default. In Fig. 14, I show the cameraman image, its Gegenbauer compressed image of a new basis, and a 1D representation of each in frequency domain to show the image truncation.



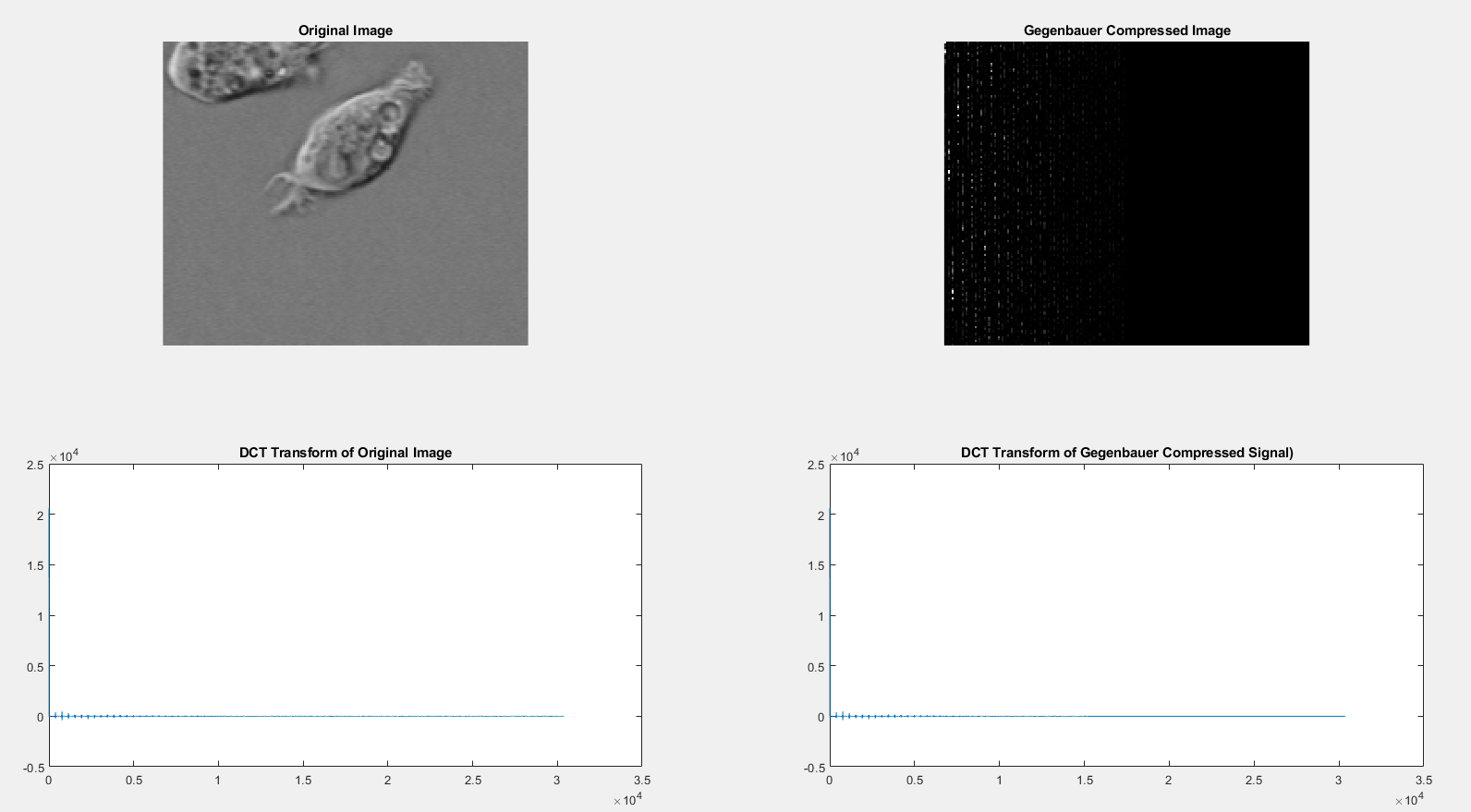


Figure 14: (a) The MatLab built-in image ‘cameraman.tif’. (b) The image (a) after Gegenbauer encoding, DCT, and truncation. (c) The 1D frequency domain of (a). (d) The 1D frequency domain of (b).

In Fig. 15, I show the cell image and its comparison the same image with the Gibbs phenomenon present and the results of IPRM on Figure 14b.

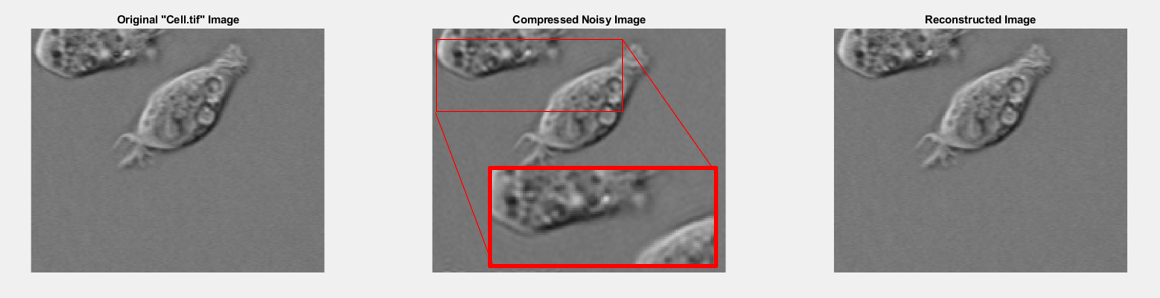


Figure 15: (a) The MatLab built-in image ‘cell.tif’. (b) The result of the image truncation shown in 14d, with the very slight Gibbs Phenomenon highlighted. (c) The image (a) after undergoing the IPRM.

I show that the IPRM reduces noise, in Fig. 16, with images at noise levels of SNR = 3dB, 10dB. and -10dB. Then I show, in Fig. 17, the root-mean-square errors of the image reconstructed without additional noise (Fig. 15c) and the image reconstructed with an SNR = -10dB (Fig. 16f).

A picture containing cat, invertebrate, gallery, different

Description automatically generated

Figure 16: (a) Fig. 14a reconstructed without IPRM at an SNR = 3dB (b) Fig. 14a reconstructed without IPRM at an SNR = 10dB (c) Fig. 14a reconstructed without IPRM at an SNR = -10dB (d) Fig. 14a reconstructed with IPRM at an SNR = 3dB (e) Fig. 14a reconstructed with IPRM at an SNR = 10dB (f) Fig. 14a reconstructed with IPRM at an SNR = -10dB

Chart, histogram

Description automatically generated

Figure 17: (a) Root-mean-square error of Fig. 14a and Fig 15c. (b) Root-mean-square error of Fig. 14a and Fig 16f.

For the third image, I consider the ‘manid.tif’ image which is built into MatLab by default. In Fig. 18, I show the cameraman image, its Gegenbauer compressed image of a new basis, and a 1D representation of each in frequency domain to show the image truncation.

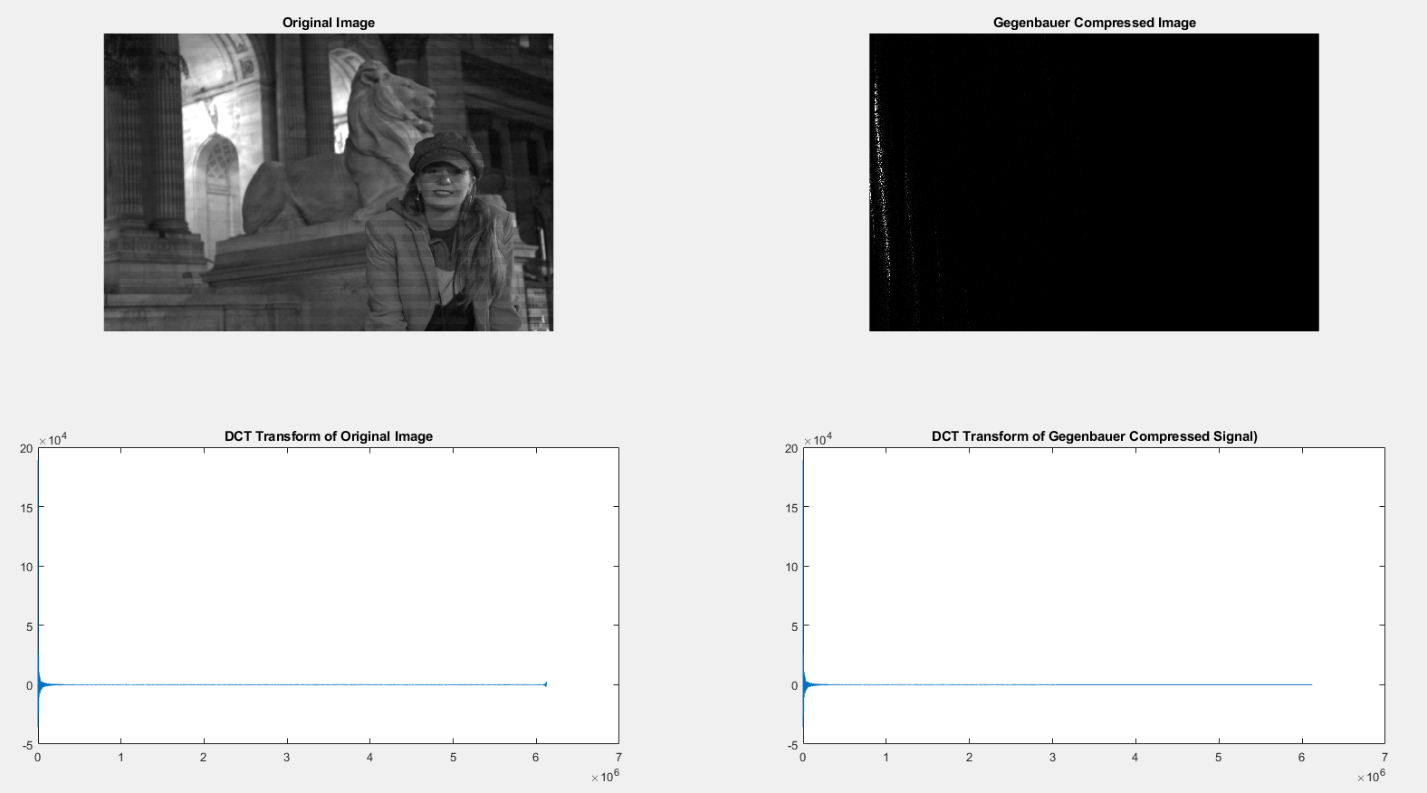


Figure 18: (a) The MatLab built-in image ‘cameraman.tif’. (b) The image (a) after Gegenbauer encoding, DCT, and truncation. (c) The 1D frequency domain of (a). (d) The 1D frequency domain of (b).

In Fig. 19, I show the Mandi image and its comparison the same image with the Gibbs phenomenon present and the results of IPRM on Figure 19b.

Graphical user interface

Description automatically generated

Figure 19: (a) The MatLab built-in image ‘mandi.tif’. (b) The result of the image truncation shown in 18d. (c) The image (a) after undergoing the IPRM.

I show that the IPRM reduces noise, in Fig. 20, with images at noise levels of SNR = 3dB, 10dB. and -10dB. Then I show, in Fig. 21, the root-mean-square errors of the image reconstructed without additional noise (Fig. 19c) and the image reconstructed with an SNR = -10dB (Fig. 12f).

A collage of people in military uniforms

Description automatically generated with low confidence

Figure 20: (a) Fig. 18a reconstructed without IPRM at an SNR = 3dB (b) Fig. 18a reconstructed without IPRM at an SNR = 10dB (c) Fig. 18a reconstructed without IPRM at an SNR = -10dB (d) Fig. 18a reconstructed with IPRM at an SNR = 3dB (e) Fig. 18a reconstructed with IPRM at an SNR = 10dB (f) Fig. 18a reconstructed with IPRM at an SNR = -10dB

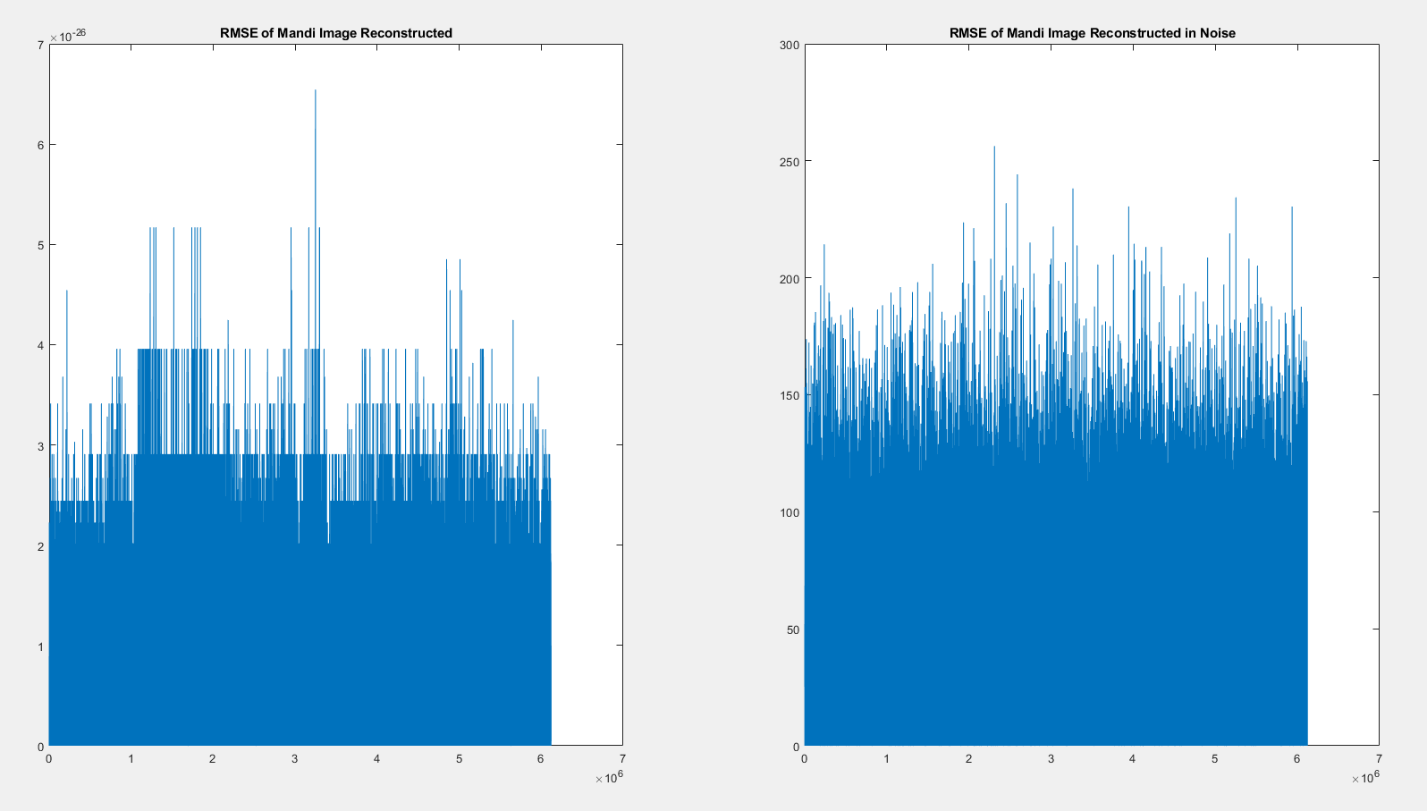


Figure 21: (a) Root-mean-square error of Fig. 18a and Fig 19c. (b) Root-mean-square error of Fig. 18a and Fig 20f.

Likewise, the next image considered is the ‘moon.tif’ image which is built into MatLab by default. In Fig. 22, I show the cameraman image, its Gegenbauer compressed image of a new basis, and a 1D representation of each in frequency domain to show the image truncation.

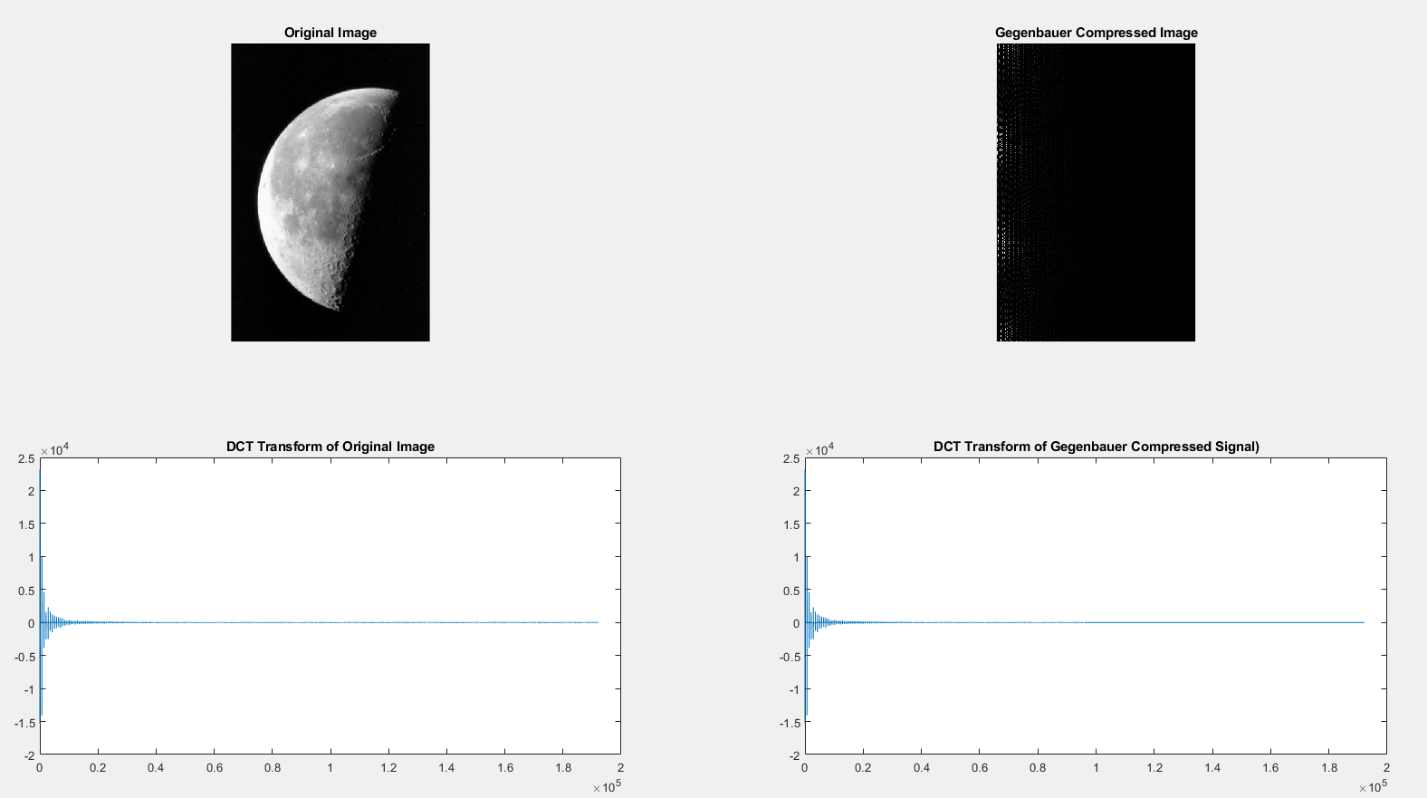


Figure 22: (a) The MatLab built-in image ‘cameraman.tif’. (b) The image (a) after Gegenbauer encoding, DCT, and truncation. (c) The 1D frequency domain of (a). (d) The 1D frequency domain of (b).

In Fig. 23, I show the moon image and its comparison the same image with the Gibbs phenomenon present and the results of IPRM on Figure 23b.

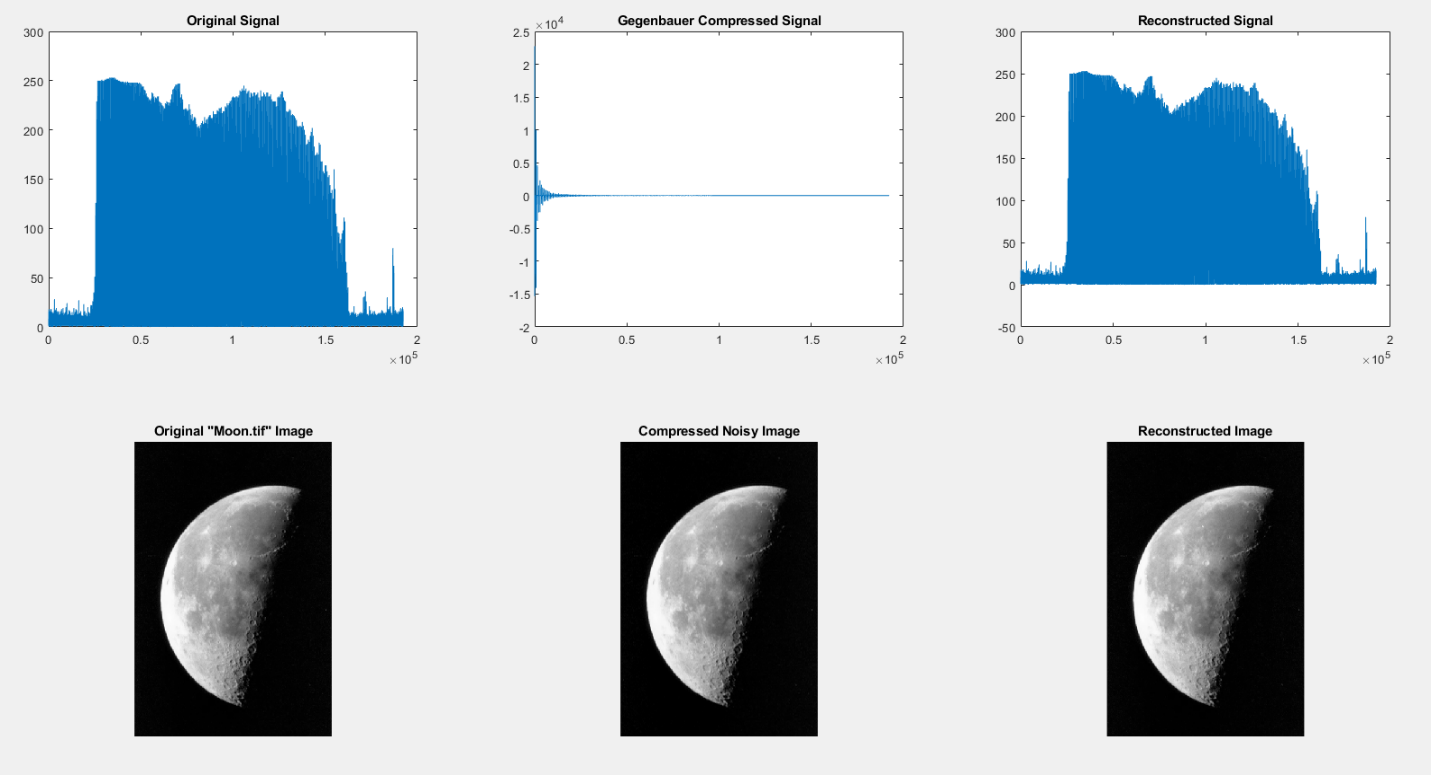


Figure 23: (a) The MatLab built-in image ‘moon.tif’. (b) The result of the image truncation shown in 22d. (c) The image (a) after undergoing the IPRM.

I show that the IPRM reduces noise, in Fig. 24, with images at noise levels of SNR = 3dB, 10dB. and -10dB. Then I show, in Fig. 25, the root-mean-square errors of the image reconstructed without additional noise (Fig. 23c) and the image reconstructed with an SNR = -10dB (Fig. 24f).

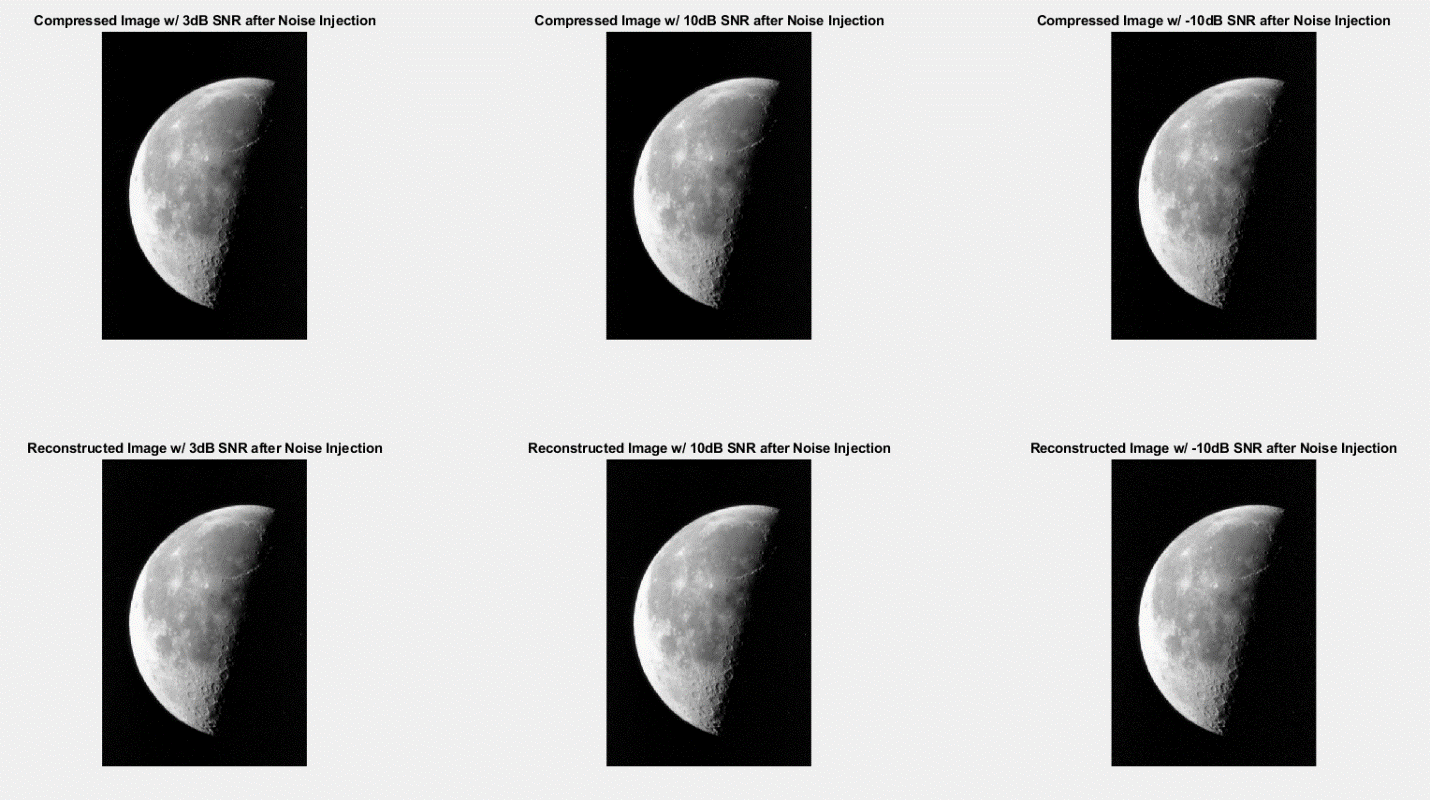
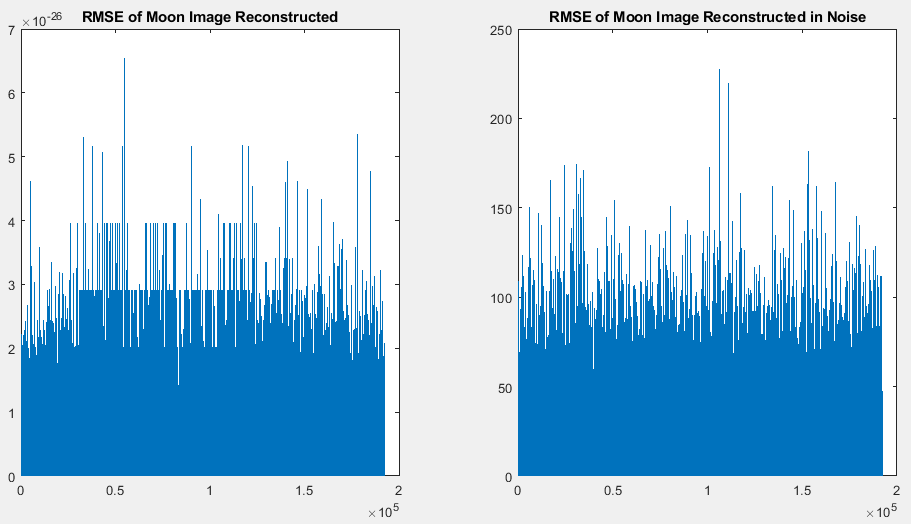


Figure 24: (a) Fig. 22a reconstructed without IPRM at an SNR = 3dB (b) Fig. 22a reconstructed without IPRM at an SNR = 10dB (c) Fig. 22a reconstructed without IPRM at an SNR = -10dB (d) Fig. 22a reconstructed with IPRM at an SNR = 3dB (e) Fig. 22a reconstructed with IPRM at an SNR = 10dB (f) Fig. 22a reconstructed with IPRM at an SNR = -10dB

Figure 25: (a) Root-mean-square error of Fig. 22a and Fig 23c. (b) Root-mean-square error of Fig. 22a and Fig 24f.

The final image considered is the ‘pout.tif’ image which is built into MatLab by default. In Fig. 26, I show the cameraman image, its Gegenbauer compressed image of a new basis, and a 1D representation of each in frequency domain to show the image truncation.

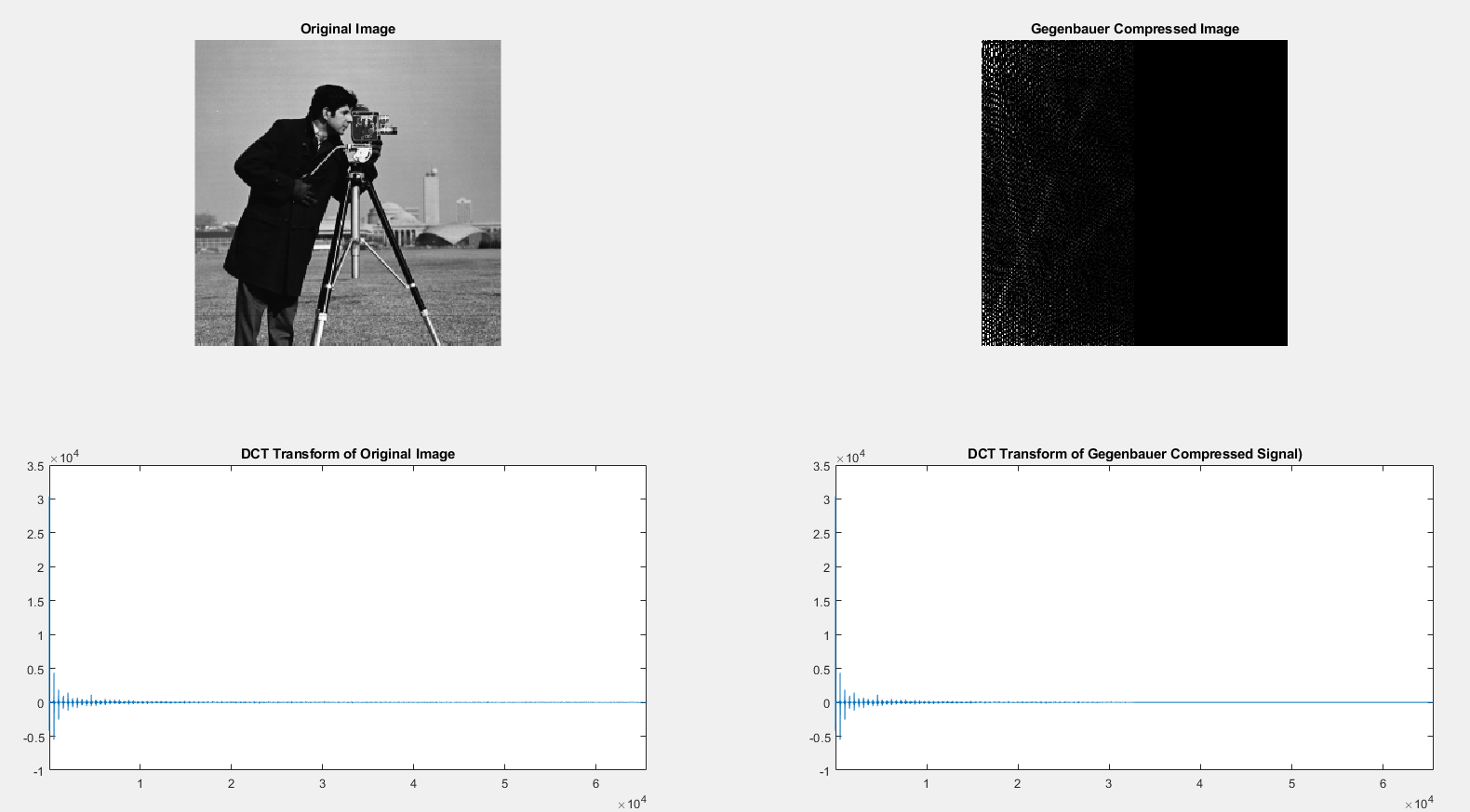


Figure 26: (a) The MatLab built-in image ‘pout.tif’. (b) The image (a) after Gegenbauer encoding, DCT, and truncation. (c) The 1D frequency domain of (a). (d) The 1D frequency domain of (b).

In Fig. 27, I show the moon image and its comparison the same image with the Gibbs phenomenon present and the results of IPRM on Figure 27b.

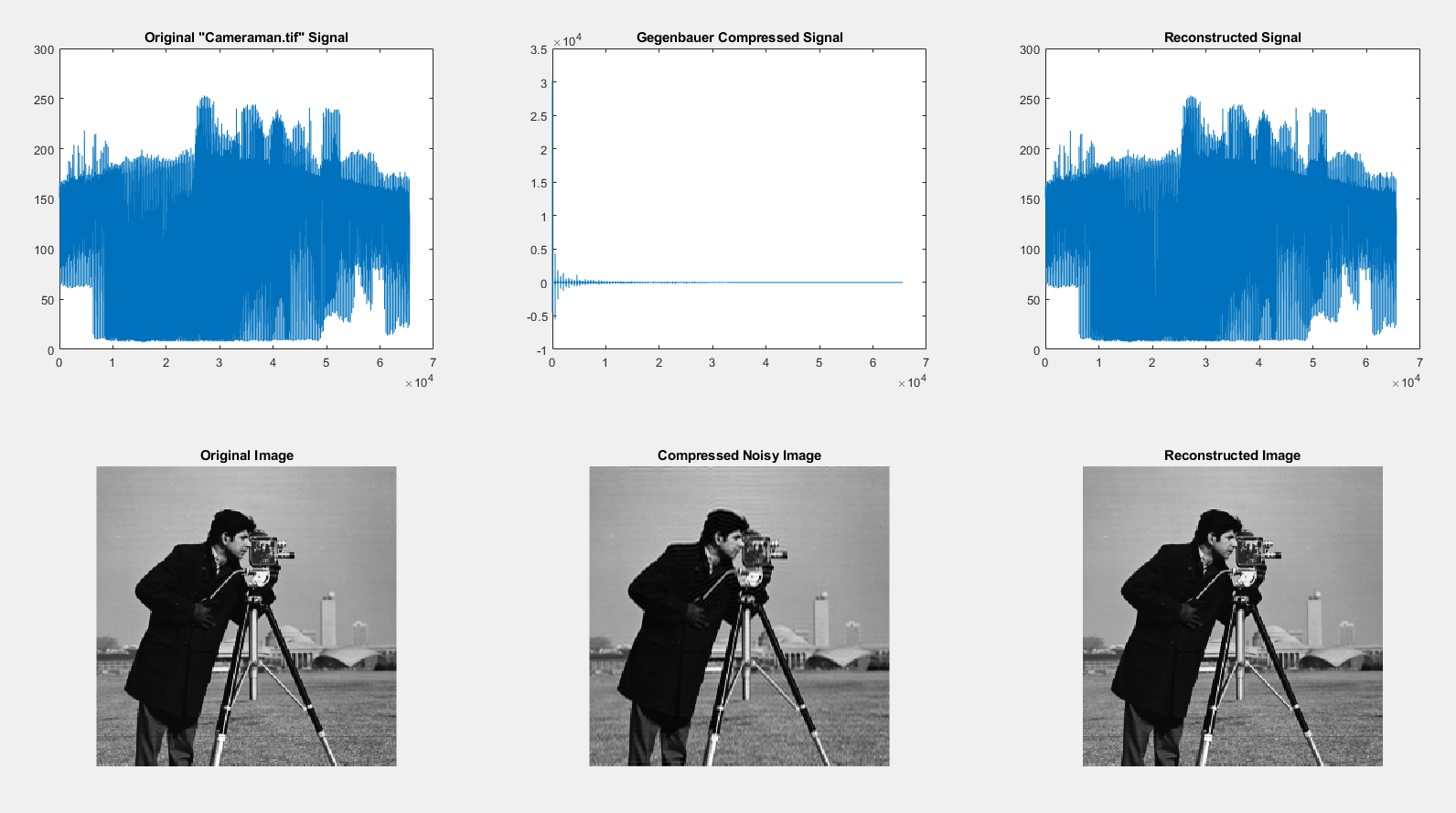


Figure 27: (a) The MatLab built-in image ‘cameraman.tif’. (b) The result of the image truncation shown in 10d. (c) The image (a) after undergoing the IPRM.

I show that the IPRM reduces noise, in Fig. 28, with images at noise levels of SNR = 3dB, 10dB. and -10dB. Then I show, in Fig. 29, the root-mean-square errors of the image reconstructed without additional noise (Fig. 27c) and the image reconstructed with an SNR = -10dB (Fig. 28f).



Figure 28: (a) Fig. 26a reconstructed without IPRM at an SNR = 3dB (b) Fig. 26a reconstructed without IPRM at an SNR = 10dB (c) Fig. 26a reconstructed without IPRM at an SNR = -10dB (d) Fig. 26a reconstructed with IPRM at an SNR = 3dB (e) Fig. 26a reconstructed with IPRM at an SNR = 10dB (f) Fig. 26a reconstructed with IPRM at an SNR = -10dB

Chart, histogram

Description automatically generatedFigure 29: (a) Root-mean-square error of Fig. 26a and Fig 27c. (b) Root-mean-square error of Fig. 26a and Fig 28f.

**5 Discussion of Results**

The

**8 References**

1. Dadkhahi et al.: Inverse polynomial reconstruction method in DCT domain. EURASIP Journal on Advances in Signal Processing 2012 2012:133.
2. Walter, G.G., Shim, HT. Gibbs' phenomenon for sampling series and what to do about it. *The Journal of Fourier Analysis and Applications* **4,**357–375 (1998). <https://doi.org/10.1007/BF02476033>
3. J. H. Jung, B. D. Shizgal. Inverse Polynomial Reconstruction of Two Dimensional Fourier Images. Journal of Scientific Computing. Vol 25, No 3. December 2005. DOI: 10.1007/s10915-004-4795-3.
4. S. A. Sarra. The MatLab Postprocessing Toolkit. Marshall University. July 2008.
5. [Weisstein, Eric W.](https://mathworld.wolfram.com/about/author.html) "Gegenbauer Polynomial." From [*MathWorld*](https://mathworld.wolfram.com/)--A Wolfram Web Resource. <https://mathworld.wolfram.com/GegenbauerPolynomial.html>
6. D. Gottlieb, C.-.W. Shu, A. Solomonoff, H. Vandeven. On the Gibbs phenomenon I: recovering exponential accuracy from the Fourier partial sum of a nonperiodic analytic function. J. Comput. Appl. Math., 43 (1992), pp. 81-98
7. Kim, D.S., Kim, T. & Rim, SH. Some identities involving Gegenbauer polynomials. *Adv Differ Equ* 2012, 219 (2012). https://doi.org/10.1186/1687-1847-2012-219
8. Parry Y. Chen, Yonatan Sivan, Resolving the Gibbs phenomenon via a discontinuous basis in a mode solver for open optical systems,Journal of Computational Physics, Volume 429, 2021, 110004, ISSN 0021-9991, <https://doi.org/10.1016/j.jcp.2020.110004>.
9. BD Shizgal, J-H Jung, Towards the resolution of the Gibbs phenomena, J. Comput. Appl. Math. 161(1), 41–65 (2003)
10. Johansson, Fredrik. “Orthogonal Polynomials.” From *mpmath*. February 2021. https://mpmath.org/doc/current/functions/orthogonal.html