

The INNOTOOL: An Image Processing Education Tool

Diego F. Pava

Florida Atlantic University, Boca Raton, Florida, USA, dpava@fau.edu

Faculty Mentors:

William T. Rhodes PhD.

Florida Atlantic University, Boca Raton, Florida, USA, wrhodes@fau.edu

Nurgun Erdol PhD.

Florida Atlantic University, Boca Raton, Florida, USA, erdol@fau.edu

ABSTRACT

Image processing is a rapidly evolving field with applications in many different disciplines. As computers can process more data in less time, algorithms increasing in complexity are being implemented to extract information from an image, make an image more readable for a human user or manipulate the image in order to enhance certain features. Innotool is a MATLAB based, graphical user interface tool for teaching image processing techniques to students of courses where image processing is important but is not the main focus of the class. The main objective is to have a tool that combines the versatility of the image processing toolbox without the learning curve associated with familiarizing the user with the language and data handling of MATLAB. The tool was part of an NSF initiative for online education in medical imaging and was tested in a medical imaging course where it obtained good feedback from the students. Innotool is divided in topical modules with the possibility of new modules to be added in the future, even modules created by third party users.

Keywords: Image Processing, MATLAB, GUI, Educational Tool.

RESUMEN

El procesamiento de imágenes es un campo que ha estado evolucionando rápidamente con aplicaciones en muchas diversas disciplinas. Con los computadores procesando una cada vez mayor cantidad de datos en menos tiempo, algoritmos cada vez más complejos están siendo implementados para extraer información de una imagen, hacer la imagen más inteligible para un usuario humano o manipular la imagen para resaltar ciertas características. Innotool es una herramienta interfaz gráfica creada en MATLAB para enseñar procesamiento de imágenes a estudiantes de cursos donde el procesamiento de imágenes es importante, pero no es el foco central de la clase. El objetivo principal es tener una herramienta que combine la versatilidad del image processing toolbox sin la curva de aprendizaje asociada con la familiarización del usuario con el lenguaje y el manejo de datos de MATLAB. La herramienta hizo parte de una iniciativa del NSF para educación en línea en imágenes médicas y fue utilizada en un curso de imágenes médicas donde obtuvo Buena retroalimentación por parte de los estudiantes. Innotool está dividida en módulos que cubren diferentes temas con la posibilidad de nuevos módulos en el futuro, incluso módulos creados por terceras personas.

Palabras claves: Procesamiento de Imágenes, MATLAB, interfaz gráfica, herramienta educacional.

1. INTRODUCTION

Human beings are primarily visual creatures, that is, they rely heavily on information coming through their eyes in order to process events. Visual information represents 99% or more of the total information received from the surroundings (Ross 1995). "An image is worth a thousand words" is as factual today as it was thousands of years ago when prehistoric men drew what they saw inside their caves.

Technological progress has naturally developed under this information bias as more scientific instruments commonly produce images to communicate results to the user or operator rather than numerical or written data that is usually harder to read and analyze.

One of the problems with this approach is that many times the images obtained are not clear enough and the information is obscured by undesired data. The obvious solution to this problem is to take another image of the same phenomena but sometimes this is not possible due to costs, the event not being repeatable or possible harm to the subject or the object being tested (Gonzalez and Woods, 2002).

The computer revolution brought with it a possible solution to this problem; a series of techniques and transformations that can be applied to the images in order to make them more readable either for humans or for computer data extraction. These series of techniques and transformations are commonly known as *image processing*, a topic that has been gaining more importance and evolving, as computers process larger amounts of data in less time.

2. MOTIVATION

Teaching *image processing* in a merely theoretical way is not an easy task. Many of the techniques being taught are better understood if the students can use them in practice and see the results for themselves. However, most of the image processing software available out there is very expensive, very limited to a specific field or more geared towards image edition than processing.

MATLAB, and specifically its Image Processing Toolbox, has proven to be a versatile, powerful and useful software for many basic and advanced image processing applications. The main advantage of MATLAB is that is a program optimized to work with and manipulate matrices, which is a big advantage for manipulating digital images over programs that work essentially with scalar values such as C. MATLAB provides not only image processing functions ready to use but the opportunity to create new functions and scripts that fit the needs of the researcher. Many image processing courses today combine the theory with the practice using MATLAB based exercises which has proven to be a very effective method (Gonzalez et al, 2004)

Image processing is a fascinating subject. Often, however, in order to appreciate what it can do, the student has to become an advanced user of the software employed for the processing of images. MATLAB programming has a steep learning curve and requires time and dedication that sometimes is too cumbersome for students who just want to know about image processing without spending a lot of time learning a new language.; especially if the user has not prior experience with the software and the class is not entirely devoted to image processing but some basic background in image processing is necessary as is the case in courses such as cognitive psychology, computer vision or medical imaging

When the National Science Foundation and the University of Miami approached us with the idea of developing a tool for teaching image processing for medical imaging students (NSF, 2010), MATLAB seemed like a natural fit. However, to really learn how to use it would take almost a whole course and would take the focus away from the main topics such as X-rays, ultrasound, MRI imaging, etc. It was clear there was a need for a tool that could combine the versatility and capabilities of MATLAB without requiring any prior MATLAB experience for the student.

3. GENERAL DESCRIPTION

The Innotool was an idea born in a creativity and innovation class (hence the name) where the challenge was to come up with a new way to teach a selected subject more easily than by means of traditional methods. Innotool was developed to offer an alternative; by means of a graphical user interface, to traditional MATLAB image processing toolbox tutorials.

Innotool uses a familiar environment of menus and buttons that perform the different tasks automatically, thereby allowing the user to learn image processing without prior knowledge of MATLAB. Innotool enables the student

to load and manipulate images using different, well-established image processing techniques without having to write a MATLAB script or function.

3.1 THE INNOTOOL ENVIRONMENT

Innotool was created using the graphical user interface capability of MATLAB. It consists of several modules, each one of them devoted to a specific topic of image processing. The main motivation to organize the program in modules is to permit the continuous evolution of the software by allowing the creation of new modules with new capabilities, even by the users.

Each module is divided in panels that cluster together buttons and numerical input that perform functions and image processing techniques that are related as a subtopic of the general topic of the module. In addition, a panel for general image transformation and a Save Image button is always present across all modules.

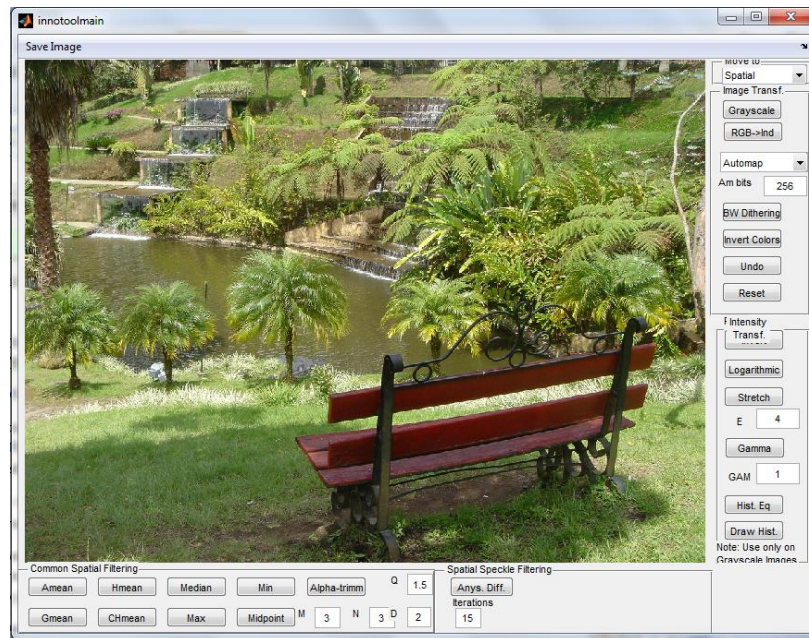


Figure 1: The Innotool

3.2 CAPABILITIES AND LIMITATIONS OF THE CURRENT VERSION

Innotool is designed to work efficiently under MATLAB version 2011b or later and with an Image Processing Toolbox version 7.3 or later. Most versions of MATLAB contain the Image Processing Toolbox including the student version.

Innotool can load images saved in most of the popular image formats such as bmp, gif, jpg, tiff, png, etc. However, the current version of the Innotool only can save images as jpg after performing the image processing. This is a limitation that will be corrected in future versions of the program. The user can select the folder in the computer where they want to store the processed image.

The Spatial Domain Module is currently set as the default module that opens when you load an image. The user can change modules at any time and combine processing techniques from different modules. The process of moving an image between modules is transparent to the user. If the user does not like the last process performed over the image, there is an undo button that will restore the image to the state it had before performing the last process, and a reset button to restore the image to its original form.

The images loaded by the user can be true color images, color indexed, grayscale or binary; for more information on each type of image please refer to (Fortner and Meyer, 1997). However, it is important to note that many of the image processing techniques requires the image to be in grayscale form and in that case, the program will

automatically convert the image from its current form to grayscale. Additionally, a grayscale button is also available in the image transformation panel and can be used at any time.

A one-time installation process of the program is necessary but once it has been installed it can be easily called from the MATLAB command window just by typing `innotool` and pressing enter. A graphical user interface should then open prompting the user to load an image.

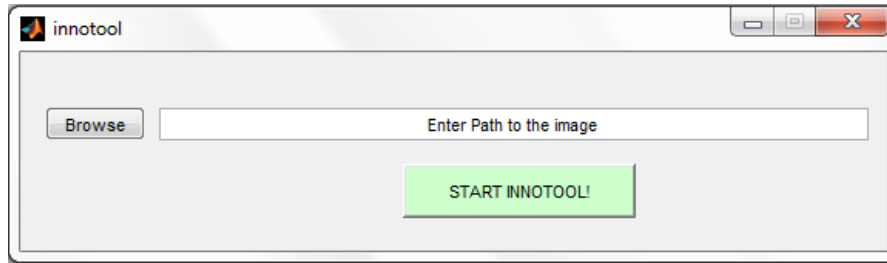


Figure 2: Initial graphical user interface

4. THE INNOTOOL MODULES

As stated above, each one of the modules consists of a series of functions and algorithms around a specific topic. The functions are clustered together in subtopics and each subtopic belongs to a panel inside the module. For the current version of Innotool there are 3 finished modules: spatial domain processing, frequency domain processing and noise models module.

4.1 THE SPATIAL DOMAIN MODULE

In Image processing language, Spatial Domain refers to the image plane itself, that is, the matrix of pixels that forms the image (Gonzalez et al, 2004). Many image modification techniques are implemented by directly working on the matrix, i.e., in the spatial domain and are thus known as Spatial Domain Image Processing Techniques.

The Spatial Domain Module brings together the most basic and most used spatial domain techniques for medical imaging available, allowing the student to directly work in and transform the image. As explained before, all of these techniques are performed on grayscale images, not in a color domain.

Mathematically, a spatial domain transformation can be expressed as $g(x,y)=T\{f(x,y)\}$, where $f(x,y)$ is the original image, $g(x,y)$ is the output image, and T is an operator working on the image $f(x,y)$. Spatial domain image processing can be divided into two sub-categories, intensity transformation functions and spatial filtering.

For intensity transformation functions, the operator T requires only the value of the pixel being analyzed in order to obtain the value of the output pixel. This means that each pixel is analyzed independently from the other pixels and so only the intensity of the pixel is of importance.

On the other hand, in spatial filtering, T operates on what is called a neighborhood around a point in $f(x,y)$. This neighborhood is usually defined as the group of pixels surrounding the point. The size and form of the neighborhood depends on the operator, but in the simplest cases are the so-called Moore and the Von Neumann neighborhoods (Griswold, 2002), which are illustrated in figure 3.

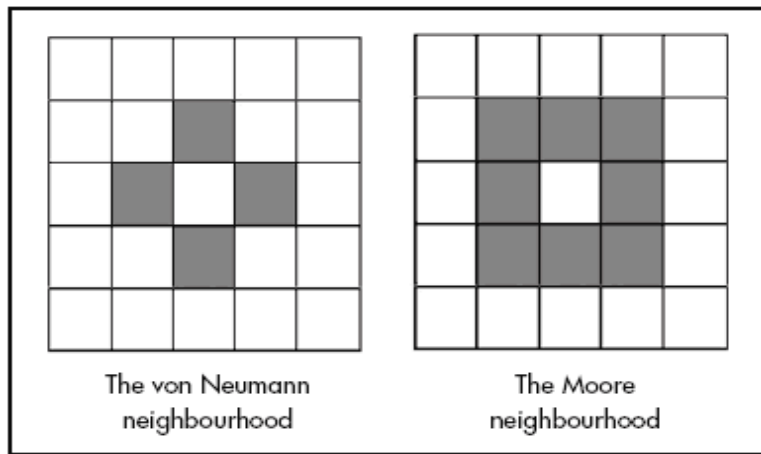


Figure 3: Initial graphical user interface

4.1.1 INTENSITY TRANSFORMATION PANEL

The intensity transformation panel allows the user to invert the image, which is the equivalent to obtain the negative of the original grayscale image. It also has buttons to modify the dynamic range of values of an image such as logarithmic, stretch and gamma corrections. All of these techniques allow to enhance certain regions of the images while compressing the less important ones.

The Intensity Transformation panel also allows us to draw the histogram of an image which is a graphic depicting how many pixels per value are there in the image. To do so, it transforms the image to a grayscale image of 255 levels of gray and then counts how many pixels are there on each level.

Lastly, the intensity transformation panel also allows us to do a histogram equalization. Let us assume that the intensity levels of an image are given by a continuous variable in the range from 0 to 1, and let us formulate the transformation given by:

$$s = T(r) = \int_0^r p_r(w) dw$$

where s is the output intensity, r is the input intensity, p_r is the probability density function of the intensity levels, and w is a dummy integration variable. It can be shown (Gonzalez et al 2004) that the output intensity will have a uniform probability density function. This means that the image would use the full range of intensities available.

This transformation will enhance the contrast of the image since all the intensities are used equally. That is why this transformation is known as an intensity equalization transformation. In reality there is not a continuous range of intensities available, and we can only work with the histogram of the image instead of the probability, in which case the transformation given by

$$s_k = T(r_k) = \sum_{j=1}^k \frac{n_j}{n}$$

will produce an image whose intensity histogram, although not uniform as in the continuous case, will have enhanced contrast and will better use of the dynamic range of intensities,

Histogram equalization will enhance contrast by redistributing the intensity of each of the pixels, the result is an image that even though does not look as “natural” it highlights the fine differences between pixel intensity levels that would be very hard to see otherwise.

The enhancement effect of the histogram equalization can be seen in figure 4.

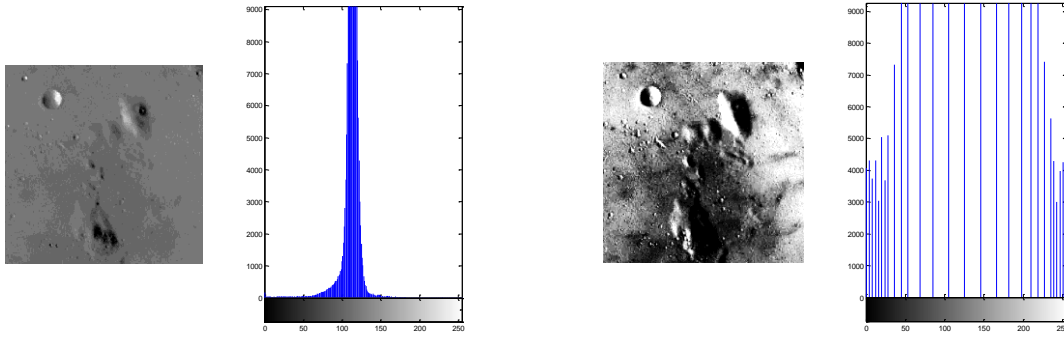


Figure 4: Original Image vs. Histogram equalized image with their respective histograms.

4.1.2 THE SPATIAL FILTERING PANEL

The first step in spatial filtering is to choose a center point and then the shape of the neighborhood. In the case of the Innotool, the shape of the neighborhood is a rectangle of size M by N that can be modified by the user just prior to pressing any of the filter buttons. The default values for both M and N is 3, corresponding to a Moore neighborhood but the user can modify them to create new neighborhoods.

The filters available in the panel are the arithmetic mean, geometric mean, harmonic mean, contraharmonic mean, median, maximum point, minimum point, mid point and alpha trimmed mean. All of them explained in detail in (Gonzalez and Woods, 2002) and (Gonzales et all, 2004).

4.2 THE SPECTRAL DOMAIN MODULE

Images can be modeled as two-dimensional functions of the form $f(x,y)$. The Fourier transform of such functions is an extension of its one dimensional counterpart with the equations for the transform and its inverse given by:

$$G(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y) \exp[-j2\pi(ux + vy)] dx dy .$$

$$g(x, y) = \mathcal{F}^{-1} \{G(u, v)\} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} G(u, v) \exp[j2\pi(ux + vy)] dudv$$

Where the variables u and v are the frequency counterparts of x and y respectively.

In image processing, the independent variables are x and y, which are spatial coordinates, and u and v are spatial frequency variables. In much the same way that a time signal is composed of various frequencies, an image is composed of various spatial frequencies. In the same way one can determine by spectral (Fourier) analysis if a temporal signal has low and high frequency components, it is also possible to determine the frequency components of an image. Regions of the image with abrupt changes in intensity, such as borders of objects, have high spatial frequency components, whereas smooth regions are mainly composed of low frequencies. Two-dimensional sinusoids of different frequencies can be seen in the figure 5.

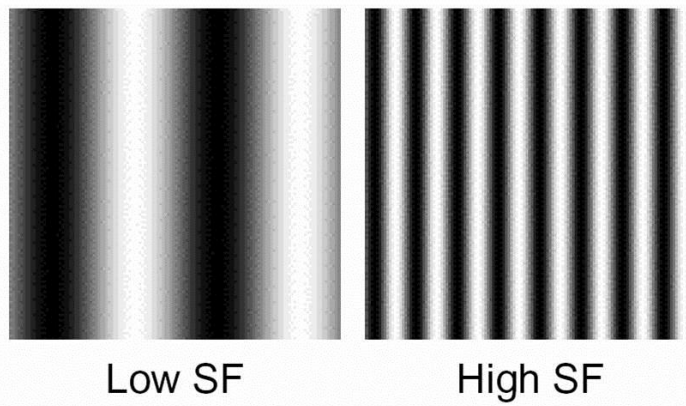


Figure 5: Sinusoidal signal with low vs high spatial frequency.

4.2.1 FOURIER TRANSFORM PANEL

The Spectrum button calculates and presents the Fourier transform of an image in graphic form. What is displayed is the spectrum, that is, the absolute value of $F(u,v)$. The drop-down menu lets the user choose between displaying the results in lineal or logarithmic form, logarithmic form usually showing more details about the spectrum.

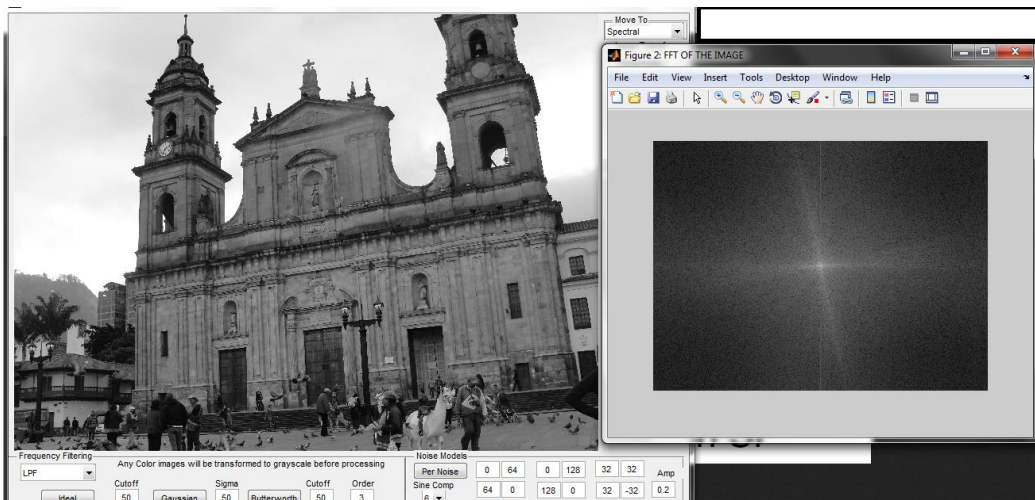


Figure 6: The Fourier Transform of an image.

4.2.2 FOURIER TRANSFORM PANEL

Filtering in the frequency domain is analogous to filtering in the spatial domain for temporal signals. At the core of the filtering processing lies the convolution theorem, for which two dimensional spatial functions can be written as:

$$f(x, y) ** h(x, y) \leftrightarrow F(u, v)G(u, v)$$

$$f(x, y)h(x, y) \leftrightarrow F(u, v) ** G(u, v)$$

Where the ** symbol means convolution in two dimensions.

Frequency filtering is done in order to modify the spectrum of an image. To achieve this, both the image and the filter are Fourier transformed and multiplied and then the product is inverse transformed.

The panel lets the user choose between a low-pass filter (LPF) or a high-pass filter (HPF) and between three different kinds of filters: ideal, Gaussian, or Butterworth.

LPF generally smooths the image, removing any high frequency components, and is comparable to to blurring the image. HPF, on the other hand, removes the low frequency components and leaves only the high frequencies, often at borders and transitions. HPF is often employed in connection with object segmentation.

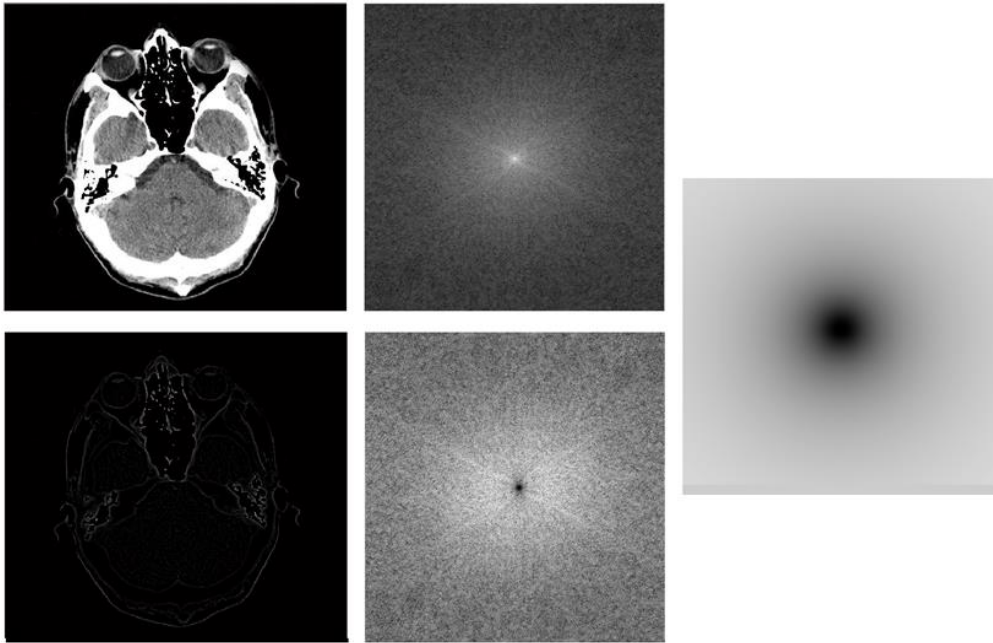


Figure 7: The CT scan when high pass filtered conserves only the borders while wiping out the low frequency regions.

4.3 THE NOISE MODULE

Images are usually imperfect versions of the original object affected by a degradation process and additive noise. Mathematically speaking, a degraded image $g(x,y)$ of an object $f(x,y)$ can be modeled as

$$g(x,y)=H[f(x,y)]+\eta(x,y)$$

The objective of image processing is to restore the image by compensating for both phenomena. The end result is an estimate of $f(x,y)$. The more the user knows about the degradation function H and the additive noise η , the closer this estimation will be to the original.

If the degradation function is linear and the same throughout the image, the equation becomes

$$g(x,y)=h(x,y)**f(x,y)+\eta(x,y)$$

$$G(u,v)=H(u,v)F(u,v)+N(x,y)$$

Noise can be understood as random fluctuations of the intensity that negatively affect the quality of the image. The source and amount of noise depend on the imaging system. Simulating the effect of noise in images is essential for the image processing learning experience. The Innotool provides the user with the ability to artificially add noise to images in order to test algorithms for noise reduction.

The noise models module has two panels. The first panel is for random spatial additive noise where the user can add different types of noise such as Gaussian, Poisson, Salt and Pepper and speckle noise to an image and set the magnitude of the noise. Refer to (Gonzalez and Woods ,2002) for explanations. The second panel is for periodic noise, which is similar to noise encountered in some uncalibrated medical imaging equipment (Prince and Links, 2005).



Figure 7: The MRI of a knee (top left). Image with Gaussian noise (top right), image with salt and pepper noise (bottom left) and image with speckle noise (bottom right)

5. THE INNOTOOL TUTORIAL

To familiarize the user with the Innotool program, a tutorial was created in pdf. form. The tutorial contains a brief introduction to each of the topics and a step by step explanation of how to download, install and use the Innotool along with several exercises and examples that are easy to follow. This tutorial, when read carefully, speeds up the learning process allowing the student to be using the program within minutes.

6. USING THE INNOTOOL IN CLASS

To Innotool was used as part of a Medical Imaging course at the Pontificia Universidad Javeriana in Bogotá, Colombia. The course is a 3 credit hours elective course for the Electronics Engineer degree program with 31 students enrolled. Most of the students had very limited experience with MATLAB and zero experience with the Image Processing Toolbox.

The students were given the task of reading the Innotool tutorial and then solve some exercises and answer some questions including a question about what they thought about the program. This task would account for a project and be worth 10% of the total grade of the class. They had one week to present the project with only a 3 hours introductory course given to them.

The results were satisfactory with a grade average of 97 out of 100 and with most of the students finding the Innotool a useful, easy to use program that required very little training time and was able to solve a variety of image processing tasks.

7. CONCLUSIONS

It is clear that Innotool has to be used in more class settings in order to definitely assess its usefulness. However, early results are encouraging in that Innotool is a program that helps students understand and use image processing techniques without the learning curve associated with familiarizing themselves with a high-end language such as MATLAB.

For an image processing centered course, it is still advisable to go the long route and teach the student the intricacies of the Image Processing Toolbox and MATLAB. But if the focus of the class is not Image Processing, but a basic understanding of the topic is necessary to complete the overall goals of the course, then Innotool is an alternative that should get consideration.

Innotool will continue to improve with more upcoming modules currently under construction such as morphological filtering, segmentation and color processing.

REFERENCES

Fortner, Brand, and Theodore E. Meyer (1997). *Number by colors: a guide to using color to understand technical data*. TELOS, Electronic Library of Science.

Gonzalez, Rafael C., and Richard E. Woods (2002), *Digital Image Processing*. 3ed: Prentice Hall Press.

Gonzalez, Rafael C., Richard E. Woods, and Steven L. Eddins (2004). *Digital image processing using MATLAB*. Pearson Education.

Griswold, Ralph (2002). "Part 1: Basic Concepts." *Automata*, Drawdown.

NSF DUE Award 1022750 (2010), http://nsf.gov/awardsearch/showAward?AWD_ID=1022750 02-20-2014

Prince, Jerry and Jonathan M. Links (2005), *Medical Imaging Signals and Systems*, Pearson Prentice Hall, (First edition);

Russ, John C.(1995), *The image processing handbook*, Chemical Rubber Corp., Boca Raton, FL.

Authorization and Disclaimer

Authors authorize LACCEI to publish the paper in the conference proceedings. Neither LACCEI nor the editors are responsible either for the content or for the implications of what is expressed in the paper.