**Extraction of Tumor from CT Brain Images and its Visualization using Contour plot in GUI**

A PROJECT REPORT

Submitted by

**Abha Pandey (10BCE0229)**

**Alisha Singla (10BCE0233)**

**Saloni Agarwal (10BCE0272)**

*in partial fulfillment for the award*

of

**B. Tech**

degree in

**Computer Science and Engineering**

**School of Computing Science and Engineering**



**May – 2014**



**School of Computing Science and Engineering**

**DECLARATION**

We hereby declare that the project entitled **“Extraction of Tumor from CT Brain Images and its Visualization using Contour plot in GUI”** submitted by us to the School of Computing Science and Engineering, VIT University, Vellore-14 in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering** is a record of bona-fide work carried out by us under the supervision of **Prof. Natarajan P, Assistant Professor (S.G.), SCSE.** We further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma of this institute or of any other institute or university.

**Abha Pandey(10BCE0229)**

**Alisha Singla(10BCE0233)**

**Saloni Agarwal(10BCE0272)**



**School of Computing Science and Engineering**

**CERTIFICATE**

The project report entitled “**Extraction of Tumor from CT Brain Images and its Visualization using Contour plot in GUI**” is prepared and submitted by **Abha Pandey (10BCE0229), Alisha Singla (10BCE0233)** and **Saloni Agarwal (10BCE0272).** Ithas been found satisfactory in terms of scope, quality and presentation as partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering** in VIT University, India.

**(Name & Signature of the Internal Guide)**

**Internal Examiner(s) External Examiner(s)**

**ACKNOWLEDGEMENT**

First of all we would like to thank our project guide, Prof. Natarajan P., Assistant Professor (S.G.) of SCSE, VIT U, for providing us his time and knowledge during the building and completion of this project. We would also like to thank our Program Chair, Prof. Hari Seetha and Dean, Dr. Venkateswaran S, for giving us this opportunity to implement our ideas and learning into shaping this project. Next we would take the opportunity to pass our thanks to our educational institute, Vellore Institute of Technology, Vellore campus, for providing ample opportunities and guidance .

**CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.no.** | **Title** | **Page number** |
|  | ***Title Page***  ***Declaration***  ***Certificate***  ***Acknowledgement***  ***Table of Contents***  ***List of Tables***  ***List of Figures***  ***List of Abbreviations***  ***Abstract*** | 1  2  3  4  5  7  8  9  10 |
| 1 | INTRODUCTION |  |
|  | 1.1 General | **11** |
|  | 1.2 Motivation | **11** |
|  | 1.3 Aim & Objective | **12** |
|  | 1.4 Related Work | **13** |
|  | 1.5 System Requirements | **14** |
| **2** | OVERVIEW OF PROPOSED SYSTEM |  |
|  | 2.1 Introduction of problem | **15** |
|  | 2.2 Gaps identified in existing system | **15** |
|  | 2.3 Proposed Solution | **16** |
| **3** | ANALYSIS AND DESIGN |  |
|  | 3.1 Requirement Analysis | **16** |
|  | 3.1.1 Functional Requirements | **17** |
|  | 3.1.2 Non-Functional Requirements | **17** |
|  | 3.2 Design of Proposed System | **18** |
|  | 3.2.1 System Architectural Design | **18** |
|  | 3.2.2 System Interface Description | **19** |
|  | 3.2.3 Considered Design Constraints | **20** |
| **4** | IMPLEMENTATION |  |
|  | 4.1 Tools Used | **21** |
|  | 4.2 Execution | **23** |
|  | 4.2.1 User Interface Design | **23** |
|  | 4.2.2 Test Cases | **24** |
| **5** | RESULTS AND DISCUSSION | **25** |
| **6** | CONCLUSION AND FUTURE ENHANCEMENTS | **32** |
| **7** | REFERENCES | **33** |
|  | APPENDIX -I | **34** |
|  | APPENDIX -II | **39** |

**LIST OF TABLES**

|  |  |
| --- | --- |
| TITLE | PAGE No. |
| System requirements  Test Cases | 14  24 |
| Table to show accuracy of algorithm | 32 |

**LIST OF FIGURES**

|  |  |
| --- | --- |
| TITLE | PAGE No. |
| System Interface Design  System Interface Description  Prototyping  Original image  Pre-processed image  Background estimation  Contrast adjustment when background is subtracted  Contrast adjustment when negative of background is subtracted  Subtraction of images to result in tumor region  Threshold image  Boundary detection  Contour of original image  Contour of tumour region  Contour of tumour boundary  Interface to select an image.  Start screen of GUI  Contour of selected image in GUI  Contourf of selected image in GUI  C-label of selected image in GUI  Interface to print current display in GUI | 18  19  23  25  25  26  26  27  27  27  27  28  28  28  29  29  30  30  31  31 |

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| **ABBREVIATIONS** | **EXPANSION** |
| CT | Computerized Tomography |
| GUI | Graphical User Interface |
| PSNR | Peak to Signal Noise Ratio |
| CPU | Central Processing Unit |
| MRI | Magneticl Resonance Imaging |

**ABSTRACT**

The field of medical imaging is gaining importance with an increase in the demand for automated, reliable, fast and efficient diagnosis which can provide insight to the image better than human eyes. Brain tumor is the second leading cause for cancer-related deaths in men in age 20 to 39 and fifth leading cause cancer among women in same age group. Brain tumors are painful and may result in various diseases if not cured properly. Diagnosis of tumor is a very important part in its treatment. Identification plays an important part in the diagnosis of benign and malignant tumors. A prime reason behind an increase in the number of cancer patients worldwide is the ignorance towards treatment of a tumor in its early stages. This paper discusses such an algorithm that can inform the user about details of tumor using basic image processing techniques. These methods include noise removal and sharpening of the image along with basic morphological functions, erosion and dilation, to obtain the background. Subtraction of background and its negative from different sets of images results in extracted tumor image. Plotting contour and c-label of the tumor and its boundary provides us with information related to the tumor that can help in a better visualization in diagnosing cases. This process helps in identifying the size, shape and position of the tumor. It helps the medical staff as well as the patient to understand the seriousness of the tumor with the help of different color-labeling for different levels of elevation . A GUI for the contour of tumor and its boundary can provide information to the medical staff on click of user choice buttons.

Keyword- Brain, CT, Contrast Adjust, Structural Element, Erosion, Dilation, Negation, Tumor Detection, Contour, C-label , GUI.

**1 Introduction**

**1.1 General**

The driving force of this project is to create a transparent environment where medical staff and patient can work in complete cooperation to achieve better results. This transparent environment will help the patient to feel secure as they will understand the treatment-process choice, which in turn will help the medical staff to handle the situation in a calm order giving them more time to think and work.

**1.2 Motivation**

A brain tumor is defined as abnormal growth of cells within the brain or central spinal canal. Some tumors can be cancerous thus they need to be detected and cured in time. The exact cause of brain tumors is not clear and neither is exact set of symptoms defined, thus, people may be suffering from it without realizing the danger. Primary brain tumors can be either malignant (contain cancer cells) or benign (do not contain cancer cells) [7].

Brain tumor oc11curred when the cells were dividing and growing abnormally. It is appear to be a solid mass when it diagnosed with diagnostic medical imaging techniques. There are two types of brain tumor which is primary brain tumor and metastatic brain tumor. Primary brain tumor is the condition when the tumor is formed in the brain and tended to stay there while the metastatic brain tumor is the tumor that is formed elsewhere in the body and spread through the brain [4].

The symptom having of brain tumor depends on the location, size and type of the tumor. It occurs when the tumor compressing the surrounding cells and gives out pressure. Besides, it is also occurs when the tumor block the fluid that flows throughout the brain. The common symptoms are having headache, nausea and vomiting, and having problem in balancing and walking. Brain tumor can be detected by the diagnostic imaging modalities such as CT scan and MRI. Both of the modalities have advantages in detecting depending on the location type and the purpose of examination needed. In this paper, we prefer to use the CT images because it is easy to examine and gives out accurate calcification and foreign mass location [4].

The CT image acquired from the CT machine give two dimension cross sectional of brain. However, the image acquired did not extract the tumor from the image. Thus, the image processing is needed to determine the severity of the tumor depends on the size [4].

The reasons for selecting CT images upon MRI images are as follows:

1. CT is much faster than MRI, making it the study of choice in cases of trauma and other acute neurological emergencies. CT can be obtained at considerably less cost than MRI.
2. CT can be obtained at considerably less cost than MRI.
3. CT is less sensitive to patient motion during the examination.
4. The imaging can be performed much more rapidly, so CT may be easier to perform in claustrophobic or very heavy patients.
5. CT can be performed at no risk to the patient with implantable medical devices, such as cardiac pacemakers, ferromagnetic vascular clips and nerve stimulators.

The focus of this project is CT brain images’ tumor extraction and its representation in simpler form such that it is understandable by everyone. Humans tend to understand colored images better than black and white images, thus, we are using colors to make the representation simpler enough to be understood by the patient along with the medical staff. Contour plot and c-label of tumor and its boundary is programmed to give 3D visualization from 2D image using different colors for different levels of intensity. A user-friendly GUI is also created which helps medical staff to attain the above objective without getting into the code.

**1.3 Aim & Objective – Problem Description**

The aim of the paper is tumor identification in brain CT images. The main reason for detection of brain tumors is to provide aid to clinical diagnosis. The aim is to provide an algorithm that guarantees the presence of a tumor by combining several procedures to provide a foolproof method of tumor detection in CT brain images. The methods utilized are filtering, contrast adjustment, negation of an image, image subtraction, erosion, dilation, threshold, and outlining of the tumor.

The focus of this project is CT brain images’ tumor extraction and its representation in simpler form such that it is understandable by everyone. Humans tend to understand colored images better than black and white images, thus, we are using colors to make the representation simpler enough to be understood by the patient along with the medical staff.

The objective of this work is to bring some useful information in simpler form in front of the users, especially for the medical staff treating the patient. Aim of this paper is to define an algorithm that will result in extracted image of the tumor from the CT brain image. The resultant image will be able to provide information like size, dimension and position of the tumor, plotting contour and c-label of the tumor and its boundary provides us with information related to the tumor that can prove useful for various cases, which will provide a better base for the staff to decide the curing procedure. Plotting contour-f plot and c-label plot of the tumor and its boundary will give easy understanding to the medical staff because humans comprehend images better with the help of different colors for different levels of intensity, giving 3D visualization from a 2D image.

**1.4 Related Work**

According to Mustaqeem, Anam, et.al, in “An efficient brain tumor detection algorithm using watershed and thresholding based segmentation” published in “International Journal 4” in 2012 [7], benign also can be growth as malignant which is consists of cancerous cells. Malignant is the rapid growing tumor which is invasive and life threatening. It is also called as brain cancer since the malignant contains cancerous cells that able to destroy any nearby cell.

The paper “Tumor Detection using Threshold operation in MRI Brain Images” by Natarajan P. et.al, [10] states that Primary brain tumors include any tumor that starts in the brain. Primary brain tumors can start from brain cells, the membranes around the brain (meninges), nerves, or glands. Tumors can directly destroy brain cells. They can also damage cells by producing inflammation, placing pressure on other parts of the brain, and increasing pressure within the skull. A metastatic brain tumor is a cancer that has spread from elsewhere in the body to the brain.

According to R. Rajeswari, P. Anadhakumar, in “Image segmentation and identification of brain tumor using FFT techniques of MRI images”, published in “ACEEE International Journal on Communication, Vol. 02” [6] the conventional definition of brain tumor includes neoplasms originating from brain parenchyma as well as from meninges and even tumors of the pituitary gland or of osseous intracranial structure that can indirectly affect brain tissues [4].

In the paper “A novel anatomical Structure segmentation method of CT head images” by X. Zang et.al[12] Histogram contains intensity value of 0-255. The zero value is the darkest part while the 255 was the white or the brightest side. Using the histogram analysis approached used the mixture Gaussian filter for the extracted part pixel intensity.

However, most of the technique used is more on MRI modality compared to CT images because it is higher resolutions. CT images of human body parts help medical doctors in diagnosing illness like brain tumor, colon cancer, lung cancer and so forth. However, it is quite difficult to obtain the important features in the images because it is limited by the image processing level and also doctor’s experience. This is expressed in “Automatic Classification and segmentation of brain tumor in CT images using optimal dominant gray level run length texture features,” by A. Padma and R. Sukanesh [13].

**1.5 System Requirements**

Table 1.5.1. System requirements

|  |  |  |  |
| --- | --- | --- | --- |
| Operating System | Processors | Disk Space | RAM |
| Windows XP Service Pack 3  Windows Server 2003 R2 with Service Pack 2  Windows Vista Service Pack 1 or 2  Windows Server 2008 Service Pack 2 or R2  Windows 7 | Any Intel or AMD x86 processor supporting SSE2 instruction se | 1 GB for MATLAB only, 3–4 GB for a typical installation | 1024 MB (At least 2048 MB recommended) |
| Mac OS X 10.5.5 (Leopard) and above  Mac OS X 10.6.x (Snow Leopard) | All Intel-based Macs | 1 GB for MATLAB only, 3–4 GB for a typical installation | 1024 MB (At least 2048 MB recommended) |
| Ubuntu 8.04, 8.10, 9.04, and 9.10  Red Hat Enterprise Linux 5.x  SUSE Linux Enterprise Desktop 11.x  Debian 5.x | Any Intel or AMD x86 processor supporting SSE2 instruction set | 1 GB for MATLAB only,  3–4 GB for a typical installation | 1024 MB (At least 2048 MB recommended) |

**2 Overview of the Proposed System**

**2.1 Introduction of problem**

The CT image acquired from the CT machine give two dimension cross sectional of brain. However, the image acquired did not extract the tumor from the image. Thus, the image processing is needed to determine the severity of the tumor depends on the size [4].

**2.2 Gaps identified in existing system**

In the existing solution of extraction of brain tumor from CT scan images tumor part is detected from the CT scan of the brain. The proposed solution also do the same thing , inform the user about details of tumor using basic image processing techniques. The methods include noise removal and sharpening of the image along with basic morphological functions, erosion and dilation, to obtain the background. Subtraction of background and its negative from different sets of images results in extracted tumor image. The difference in the proposed solution with existing solution is plotting contour and c-label of the tumor and its boundary which provides us with information related to the tumor that can help in a better visualization in diagnosing cases. This process helps in identifying the size, shape and position of the tumor. It helps the medical staff as well as the patient to understand the seriousness of the tumor with the help of different color-labeling for different levels of elevation

**2.3 Proposed solution**

The algorithm is a set of image processing fundamental procedures. A set of noise-removal functions accompanied with morphological operations that result in clear image of tumor after passing through high pass filter is the basic idea behind the proposed algorithm. The set of morphological operations used will decide the clarity and quality of the tumor image.

A GUI is created in the MATLAB offering the proposed application of extracting the tumor from selected brain image and its visualization using contour plot. Without having to deal with the code, medical staff can select the CT image and study the extracted tumor along with its boundary from contour and c-label options. The GUI also contains options for zoom-in, zoom-out, data cursor for co-ordinates, and prints the selected image.

**3 Analysis and Design**

**3.1 Requirement Analysis**

System requirements set out what the system should do without specifying how it should be done. The requirement set out in this document is complete and consistent. There are 2 types of user of this software-

1. Patient

2. Doctor

1. Patient can use the software to see the size of the tumor. It’s easy for the laymen to understand the size and position of the tumor.

2. Doctor are using for extracting of tumor from CT scan images of brain and visualization of tumor using contour plot.

**3.1.1 Functional Requirements**

* Selecting the CT scan images of brain.
* Extracting only tumor region from the scan images.
* Finding the boundary of the tumor.
* Plotting the contour of tumor and boundary of the tumor.
* Plotting the c-label of tumor and boundary of the tumor.
* Creating a GUI for easy access of the program.

**3.1.2 Non Functional Requirements**

* Availability- The software for Extraction of brain tumor from CT scan images can be available in all the systems who have MATLAB installed.
* Reliability- This software attempts to insure appropriate content but assume no responsibility for external manipulations.
* Performance- CPU time of the proposed software varies from 4seconds to 6seconds and PSNR value from 25dB to 26dB.

**3.2 Design of the proposed system**

**3.2.1 System Architectural Design**

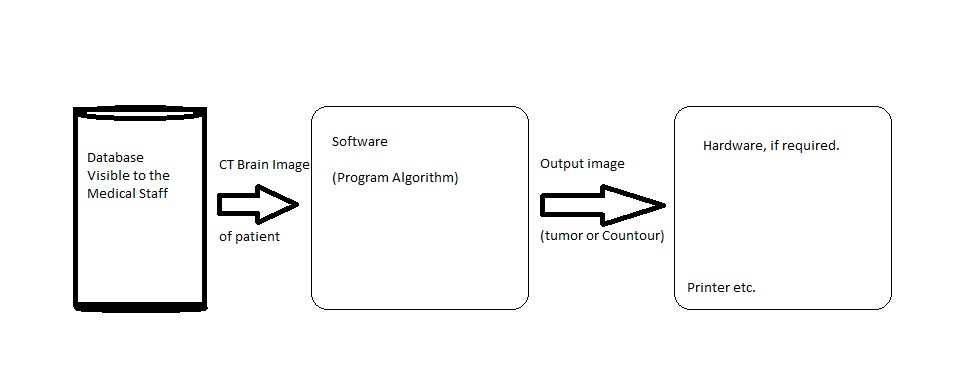


Fig 3.2.1.1.System Architectural Description

**3.2.2 System Interface Description**

Contour and C-label

CT Brain Image (with tumor)

Pre-Processing   
(Grey-scale; Noise Removal; Sharpening)

Background Estimation   
(Erosion and Dilation)

Subtraction of background and negative of background from eroded image

Contrast adjustment of resultant images.

Subtraction of boundary from contrast adjusted image

Contour and C-label

Threshold

Contour and C-label

Boundary Detection

Fig 3.2.2.1.System Interface Description

**3.2.3 Considered Design Constraints**

### 3.2.3.1 User Interface Constraints

Using this system is fairly simple and intuitive. A user familiar with basic computer operability skills should be able to understand all functionality provided by the system.

### 3.2.3.2 Hardware Constraints

The system should work on most home desktop and laptop computers and can be extended to mobile phone apps.

### 3.2.3.3 Software Constraints

The system is designed to run on MATLAB having GUIDE.

### 3.2.3.4 Communications Constraints

System must have access to the images of CT scan for brain tumour.

### 3.2.3.5 Data Management Constraints

System shall be able to interface with other components according to their specifications.

### 3.2.3.6 Operational Constraints

The system is not limited to any Operating System. It works in equally good in Windows , Mac and LINUX.

**4 Implementation**

**4.1 Tools Used**

1. **Noise Removal and Sharpening**

As a grayscale or colored image maybe the inputted image, the first step is to convert the given image into a grayscale image [4]. On procuring the grayscale image, the aim then is to filter [4] it so as to sharpen it and remove any noise [4], if present. In the algorithm, unsharp [1] [14] filtering of fspecial [2] [14] filter is applied in order to sharpen the image by removing the low intensity values. For noise-removal ‘Gaussian’ [1] [9] filters is used from fspecial filters.

1. **Erosion and Dilation**

After pre-processing, next step is to estimate the background [14]. In order to do so we make use of the basic morphological operations, erosion and dilation. More erosion [14] and less dilation [14] will result in decrease in skull bones’ image size. To accomplish this we will keep the eroding structural element’s radius bigger than that of dilating structural elements. The structuring element used is ‘diamond’.

1. **Negation**

The estimated background, obtained by the previous step, will contain the eroded tumor region as our aim was to remove the skull boundary and radius of structuring element was kept as such. Negative of the image can be calculated by subtracting [11] the image from 255 which the highest value any pixel can have.

1. **Subtraction**

Subtracting background and negative of background from eroded image will result in images with and without tumors [11]. These images will contain skull’s boundary along with the tumor region and thus will be imperfect for use [6].

1. **Contrast Adjustment**

In order to provide a clear and well-defined image to work upon, this operation is further applied to the result of subtracting images in previous step. This operation involves increasing the contrast of the filtered image, which is accomplished by performing contrast adjustment techniques [6]. These contrast images will further be subtracted from dilated image [6] [11].

1. **Threshold**

Next step in this algorithm is to calculate global image threshold using Otsu's method [10], which chooses the threshold to minimize the intra-class variance of the black and white pixels. Thus we will get a clear image of the tumor region [6].

1. **Boundary Detection**

In earlier times without aid of medical imaging tumors were identified manually and boundaries were drawn around it by an expert which always contained issues related to manual-error. Thus, to remove this error, the next step includes producing a clear boundary of the identified tumor using the morphological operation ‘remove’, which removes all the interior pixels, thus leaving only the boundary pixels on.

1. **Contour and C-label**

Contour is a curve along which the function has a constant value. A contour line (often just called a "contour") joins points of equal elevation (height) above a given level. These different levels are represented by different colored boundaries. Contour-f function gives a better view of the system by each level with different colors. C-label adds height labels to a 2-D contour plot, providing a better insight to the image.

**4.2 Execution**

**4.2.1. User Interface Design**

The prototyping of the GUI is done by pencil2.0.3 tool .

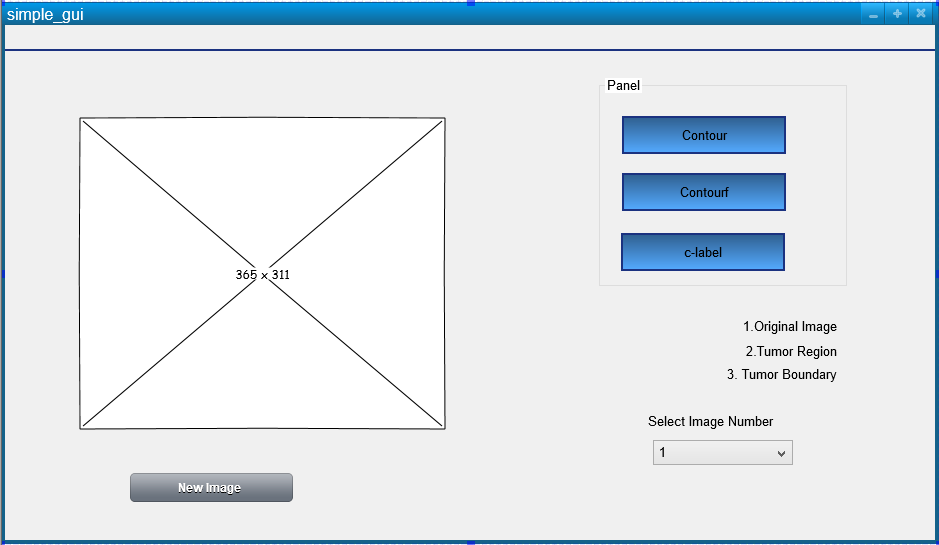
****

Fig 4.2.1.1. Prototyping

**4.2.2 Test Cases**

|  |  |  |
| --- | --- | --- |
| Initial image | Expected Output | Obtained Output |
| **E:\documents\sem 8\saloni\brain1.jpg** | E:\documents\sem 8\saloni\brain1tu.PNG | E:\documents\sem 8\saloni\brain1tu.PNG |
| **E:\documents\sem 8\saloni\brain3.jpg** | E:\documents\sem 8\saloni\brain3tu.PNG | E:\documents\sem 8\saloni\brain3tu.PNG |
| **E:\documents\sem 8\saloni\brain2.jpg** | E:\documents\sem 8\saloni\brain2tu.PNG | E:\documents\sem 8\saloni\brain2tu.PNG |
| **E:\documents\sem 8\saloni\brain4.jpg** | E:\documents\sem 8\saloni\brain4tu.PNG | E:\documents\sem 8\saloni\brain4tu.PNG |

Table 4.2.2.1. Test cases

**5 Results and Discussion**

This paper focuses upon the detection and visualization of a tumour in the brain from CT images. By developing the proposed architecture, the demarcation of the tumour in the CT image is obtained. The following results showcase the outputs received after each step in the algorithm.

After the original image undergoes pre-processing transformations we get figure 5.2 from figure 5.1. These basic pre-processing transformations include:

1. Changing the image to greyscale, as we need to find contour of the final image which works on greyscale images.
2. Applying low pass filter, to remove any noise, if present, in the image.
3. Applying high pass filter, to obtain sharpened image with clear-defined boundaries.

|  |  |
| --- | --- |
| Fig. 5.1. Original image | Fig. 5.2. Pre-processed image |
| Series of steps that lead to figure 5.3 from figure 5.2 are:   1. Eroding with structural element of big radius, say 6, to erode more. 2. Dilating with structural element of small radius, say 3, to dilate lesser than erode. 3. Use morphological open to get same amount of erosion and dilation, to estimate background | |
| Fig. 5.3. Background estimation | Fig. 5.4. Contrast adjustment when background is subtracted |
| Transitions from figure 5.3 to figure 5.4 and figure 5.5 happens by following steps mentioned below:   1. Negate background. 2. Delete background from eroded iamge. 3. Delete negative of background form eroded image. 4. Find contrast of both subtracted images. | |
| Fig. 5.5. Contrast adjustment when negative of background is subtracted | Fig. 5.6. Subtraction of images to result in tumor region |
| Subtracting figure 5 from figure 6 results in figure 7. | |
| Fig. 5.7. Threshold image | Fig.5.8. Boundary detection |
| In order to obtain figure 5.7 from figure 5.6 we take threshold of tumor region image (figure 5.7), in this process we convert the image to binary.  For boundary detection we use an inbuilt MATLAB function, bwmorph. | |
| Fig. 5.9. Contour of original image | Fig. 5.10. Contour of tumour region |
| Fig. 5.11. Contour of tumour boundary |  |
| Finally, we find contour of grey image, tumor region (figure 5.10) and tumor boundary (figure 5.11). | |

A GUI, programmed in MATLAB shows the following step by step user interfaces.

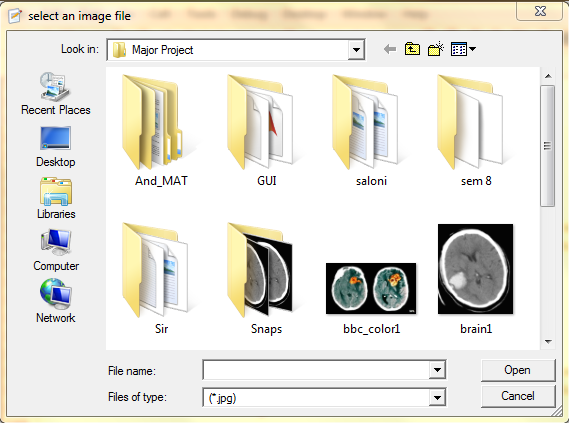


Fig 5.12. Interface to select an image.

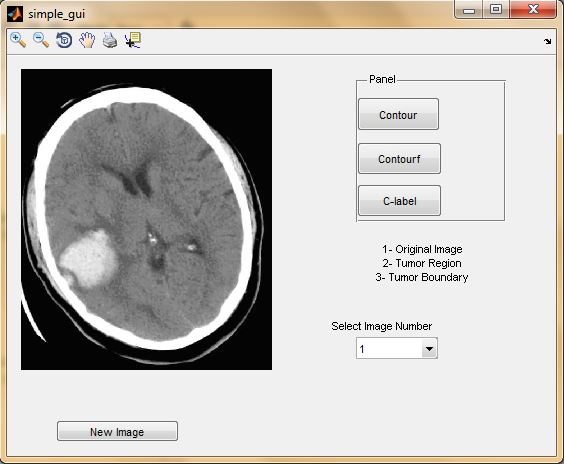


Fig 5.13. Start screen of GUI

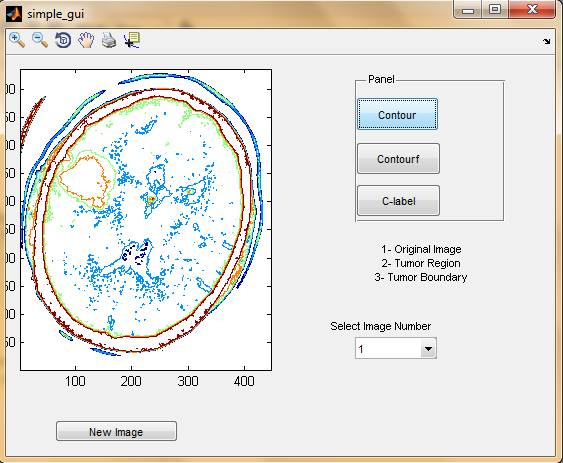


Fig 5.14. Contour of selected image in GUI

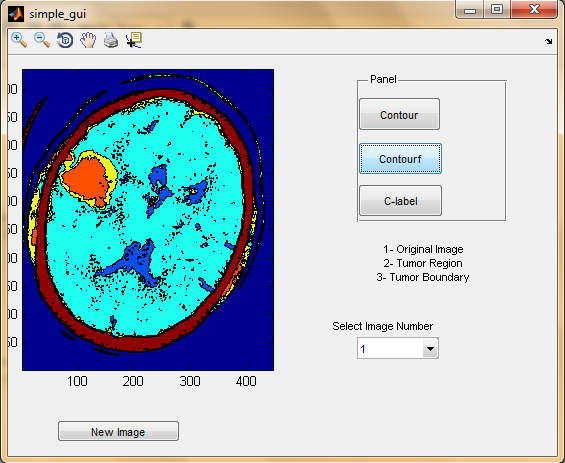


Fig 5.15. Contourf of selected image in GUI

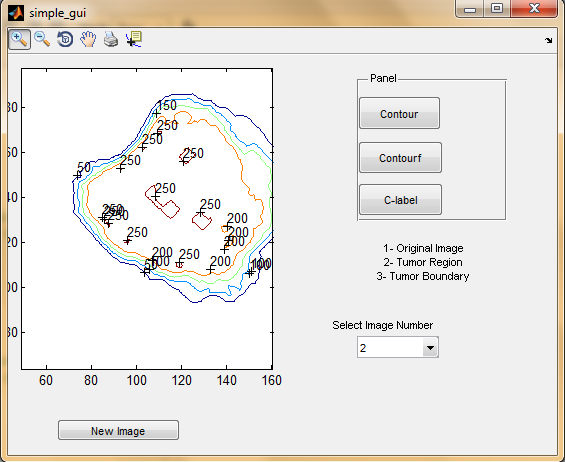


Fig 5.16.C-label of selected image in GUI

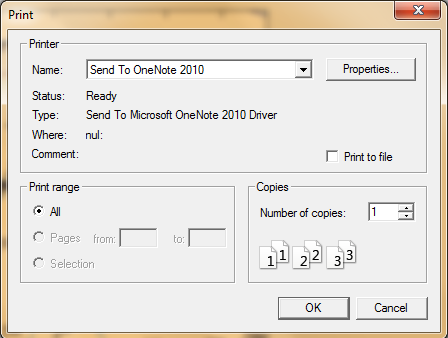


Fig 5.17. Interface to print current display in GUI

**6 Conclusion and Future Enhancements**

The proposed algorithm is inputted with gray scale images of brain that contain tumour/s. The image is processed through various stages of morphological operations like filtering, contra adjustment, erosion, dilation etc. through MATLAB programming. Hence, the tumour is outlined in the original image and clearly demarcated. Contour plot and c-label plot is created to provide 3D visualization from the 2D image. A GUI is also developed which enables the above application with a user friendly interface.

Table 6.1. Table to show accuracy of algorithm

|  |  |  |
| --- | --- | --- |
| **Image** | **CPU time (in sec)** | **PSNR** |
| **E:\documents\sem 8\saloni\brain1.jpg** | **6.2656** | **+25.84 db** |
| **E:\documents\sem 8\saloni\brain3.jpg** | **4.0156** | **+25.25 db** |
| **E:\documents\sem 8\saloni\2original.PNG** | **6.7344** | **+26.70 db** |
| **E:\documents\sem 8\saloni\brain4.jpg** | **4.7188** | **+25.73 db** |

Possible extension of the presented work could use more features. It would be beneficial to connect the system to cloud storage of patient’s information in hospital. The application can be extended to accessibility and usability through mobile phones.

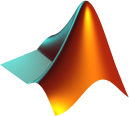
**7 References**

1. W.Gonzalez, “Digital Image Processing”, 2nd ed. Prentice Hall, Year of Publication 2008.
2. Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, Second Edition.
3. Rafael C. Gonzalez, Richard E. Woods, Steve L. Eddins, Digital Image Processing Using MATLAB, 2003.
4. Rania Hussien Al-Ashwal, Eko Supriyanto, et.al., “Digital Processing for Computed Tomography Images: Brain Tumor Extraction and Histogram Analysis”, Mathematics and Computers in Contemporary Science, 2013
5. J. Selvakumar, A. Lakshmi and T. Arivoli, “Brain Tumor Segmentation and its area calculation in Brain MR images using K-means clustering and fuzzy C-mean algorithm”, Imternational Conference on Advances in Engineering, Science and Management, 2012
6. R. Rajeswari, P. Anadhakumar, “Image segmentation and identification of brain tumor using FFT techniques of MRI images”, ACEEE International Journal on Communication, Vol. 02, No. 02, July 2011
7. Mustaqeem, Anam, Ali Javed, and Tehseen Fatima, “An efficient brain tumor detection algorithm using watershed and thresholding based segmentation”, International Journal 4, 2012
8. P.Dhanalakshmi, T.Kanimozhi, “Automated Segmentation of Brain Tumor using K-Means Clustering and its area calculation”, IJAEEE, 2013.
9. Q.Hu, G. Quian, A. Aziz, W.L.Nowinski,”Segmentation of Brain from Computed Tomography head images,” Engineering in Medicine and Biology 27th Annual Conference, 2005.
10. Natrajan P. , Krishnan N. , Natasha Sandeep kenkre and et.al ,”Tumor Detection using Threshold operation in MRI Brain Images,” IEEE International Conference on Computational Intelligence and Computing Research, 2012.
11. P. Natrajan, Debsmita Ghosh, kenkre Natasha Sandeep, Sabiha Jilani, ”Detection of Tumor in Mammogram Images using extended Local Minima Threshold,” International Journal of Engineering and Technology, Vol. 5, No . 3, jun-jul 2013.
12. X. Zang, J.Yang, D.Weng, Y. Liu and Y. Wang, “A novel anatomical Structure segmentation method of CT head images,” International Conferences on complex medical Engineering, 2010.
13. A. Padma and R. Sukanesh, “Automatic Classification and segmentation of brain tumor in CT images using optimal dominant gray level run length texture features,” International journal of Advanced Computer Science and Applications, 2011.
14. R.C. Patil and Dr. A.S. Bhalachandra, “Brain tumour extraction from MRI images using MATLAB,” International Journal of Electronics, Communication &Soft Computing Science and Engineering,vol. 2, pp. 1-4.

**APPENDIX-I**

**MATLAB**

**MATLAB** (**mat**rix **lab**oratory) is a multi-paradigm numerical computing environment and fourth-generation programming language. It uses the L-shaped membrane logo. Developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, and FORTRAN.



MATLAB® is a high-level language and interactive environment for numerical computation, visualization, and programming. Using MATLAB, we can analyze data, develop algorithms, and create models and applications. The language, tools, and built-in math functions enable us to explore multiple approaches and reach a solution faster than with spreadsheets or traditional programming languages, such as C/C++ or Java™.

Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine, allowing access to symbolic computing capabilities. An additional package, Simulink, adds graphical multi-domain simulation and Model-Based Design for dynamic and embedded systems.

In 2004, MATLAB had around one million users across industry and academia. MATLAB users come from various backgrounds of engineering, science, and economics. We can use MATLAB for a range of applications, including signal processing and communications, image and video processing, control systems, test and measurement, computational finance, and computational biology.

Cleve Moler, the chairman of the computer science department at the University of New Mexico, started developing MATLAB in the late 1970s. He designed it to give his students access to LINPACK and EISPACK without them having to learn Fortran. It soon spread to other universities and found a strong audience within the applied mathematics community. Jack Little, an engineer, was exposed to it during a visit Moler made to Stanford University in 1983. Recognizing its commercial potential, he joined with Moler and Steve Bangert. They rewrote MATLAB in C and founded MathWorks in 1984 to continue its development.

The MATLAB application is built around the MATLAB language, and most use of MATLAB involves typing MATLAB code into the Command Window (as an interactive mathematical shell), or executing text files containing MATLAB code, including scripts and/or functions.

**GUIDE**

GUIs (also known as graphical user interfaces or UIs) provide point-and-click control of software applications, eliminating the need to learn a language or type commands in order to run the application.

MATLAB apps are self-contained MATLAB programs with GUI front ends that automate a task or calculation. The GUI typically contains controls such as menus, toolbars, buttons, and sliders. Many MATLAB products, such as Curve Fitting Toolbox, Signal Processing Toolbox, and Control System Toolbox, include apps with custom user interfaces. You can also create your own custom apps, including their corresponding UIs, for others to use.

MATLAB supports developing applications with graphical user interface features. MATLAB includes GUIDE (GUI development environment) for graphically designing GUIs. It also has tightly integrated graph-plotting features. For example the function *plot* can be used to produce a graph from two vectors *x* and *y*.

To create GUI in MATLAB follow the steps listed below.

1. Click on GUIDE button on toolbar in MATLAB launch pad.
2. Make your choice of the type of GUI you need.
3. Make buttons and/or menus from the tools in GUIDE.
4. Double click on menu that is created in grey area of the GUIDE, property manager will open.
5. Change properties according to requirement and change the code if needed.

**THRESHOLD**

**Thresholding** is the simplest method of image segmentation. From a grayscale image, thresholding can be used to create binary images. Sezgin and Sankur (2004) categorize thresholding methods into the following six groups based on the information the algorithm manipulates.

1. **Histogram shape**-based methods, where, for example, the peaks, valleys and curvatures of the smoothed histogram are analyzed
2. **Clustering**-based methods, where the gray-level samples are clustered in two parts as background and foreground (object), or alternately are modeled as a mixture of two Gaussians
3. **Entropy**-based methods result in algorithms that use the entropy of the foreground and background regions, the cross-entropy between the original and binarized image, etc.
4. **Object Attribute**-based methods search a measure of similarity between the gray-level and the binarized images, such as fuzzy shape similarity, edge coincidence, etc.
5. **Spatial** methods [that] use higher-order probability distribution and/or correlation between pixels
6. **Local** methods adapt the threshold value on each pixel to the local image characteristics.

We have considered Otsu’s method of thresholding in the proposed algorithm. The algorithm assumes that the image to be thresholded contains two classes of pixels or bi-modal histogram (e.g. foreground and background) then calculates the optimum threshold separating those two classes so that their combined spread (intra-class variance) is minimal.

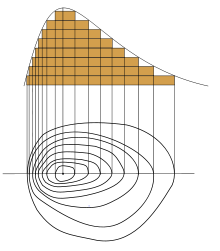
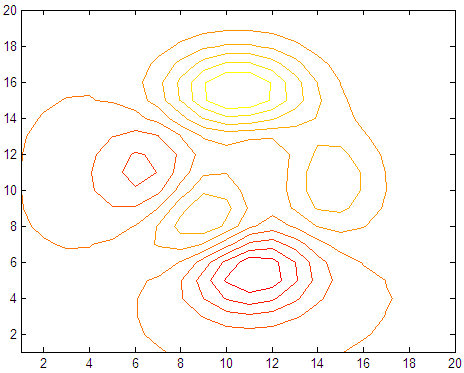
 

**CONTOUR**

A contour line (also isoline, isopleth, or isarithm) of a function of two variables is a curve along which the function has a constant value.  In cartography, a contour line (often just called a "contour") joins points of equal elevation (height) above a given level, such as mean sea level. A contour map is a map illustrated with contour lines, for example a topographic map, which thus shows valleys and hills, and the steepness of slopes. The contour interval of a contour map is the difference in elevation between successive contour lines.

More generally, a contour line for a function of two variables is a curve connecting points where the function has the same particular value. The gradient of the function is always perpendicular to the contour lines. When the lines are close together the magnitude of the gradient is large: the variation is steep. A level set is a generalization of a contour line for functions of any number of variables.

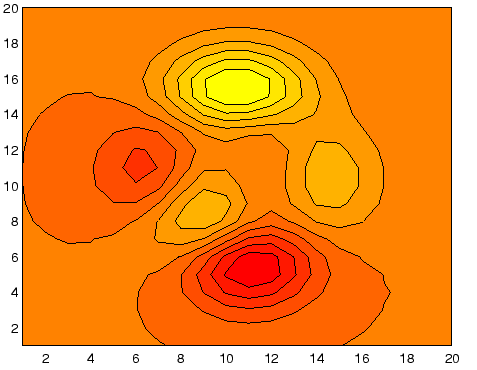
Contour lines are curved, straight or a mixture of both lines on a map describing the intersection of a real or hypothetical surface with one or more horizontal planes. The configuration of these contours allows map readers to infer relative gradient of a parameter and estimate that parameter at specific places. Contour lines may be either traced on a visible three-dimensional model of the surface, as when a photogrammetrist viewing a stereo-model plots elevation contours, or interpolated from estimated surface elevations, as when a computer program threads contours through a network of observation points of area centroids. In the latter case, the method of interpolation affects the reliability of individual isolines and their portrayal of slope, pits and peaks.

**CONTOUR-F**

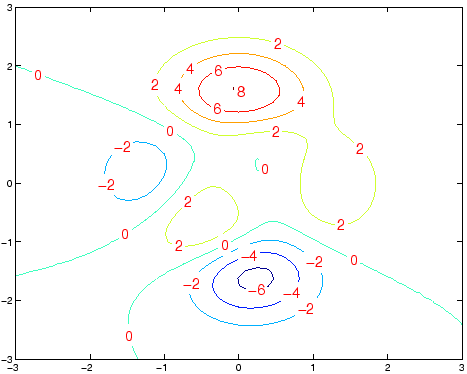
Contour-f can be explained through its name, ‘contour-filled’. A filled contour plot displays isolines calculated from matrix Z and fills the areas between the isolines using constant colors corresponding to the current figure's colormap. It is filled two-dimensional contour plot.

Contourf (Z) draws a filled contour plot of matrix Z, where Z is interpreted as heights with respect to the *x*-*y* plane.Z must be at least a 2-by-2 matrix that contains at least two different values. The number of contour lines and the values of the contour lines are chosen automatically based on the minimum and maximum values of Z. The ranges of the *x*- and *y*-axis are [1:n] and [1:m], where [m,n] = size(Z).



**C-LABEL**

C-label stands for ‘Contour plot elevation labels’. In other words, the clabel function adds height labels to a two-dimensional contour plot.



**CT IMAGES**

X-ray computed tomography (x-ray CT) is a technology that uses computer-processed x-rays to produce tomographic images(virtual 'slices') of specific areas of the scanned object, allowing the user to see what is inside it without cutting it open. Digital geometry processing is used to generate a three-dimensional image of the inside of an object from a large series of two-dimensional radiographic images taken around a single axis of rotation. Medical imaging is the most common application of x-ray CT. Its cross-sectional images are used for diagnostic and therapeutic purposes in various medical disciplines. The rest of this article discusses medical-imaging x-ray CT; industrial applications of x-ray CT are discussed at *industrial computed tomography scanning*.

As x-ray CT is the most common form of CT in medicine and various other contexts, the term computed tomography alone (or CT) is often used to refer to x-ray CT, although other types exist (such as positron emission tomography [PET] and single-photon emission computed tomography [SPECT]). Older and less preferred terms that also refer to x-ray CT are computedaxial tomography (CAT scan) and computer-assisted tomography. X-ray CT is a form of radiography, although the word "radiography" used alone usually refers, by wide convention, to non-tomographic radiography.

There are several advantages that CT has over traditional 2D medical radiography. First, CT completely eliminates the superimposition of images of structures outside the area of interest. Second, because of the inherent high-contrast resolution of CT, differences between tissues that differ in physical density by less than 1% can be distinguished. Finally, data from a single CT imaging procedure consisting of multi-planar reformatted imaging.

**APPENDIX-II**

**PSEUDO CODE**

function varargout = simple\_gui(varargin)

% SIMPLE\_GUI M-file for simple\_gui.fig

% SIMPLE\_GUI, by itself, creates a new SIMPLE\_GUI or raises the existing

% singleton\*.

% H = SIMPLE\_GUI returns the handle to a new SIMPLE\_GUI or the handle to

% the existing singleton\*.

% SIMPLE\_GUI('CALLBACK',hObject,eventData,handles,...) calls the local

% function named CALLBACK in SIMPLE\_GUI.M with the given input arguments.

% SIMPLE\_GUI('Property','Value',...) creates a new SIMPLE\_GUI or raises the

% existing singleton\*. Starting from the left, property value pairs are

% applied to the GUI before simple\_gui\_OpeningFcn gets called. An

% unrecognized property name or invalid value makes property application

% stop. All inputs are passed to simple\_gui\_OpeningFcn via varargin.

% \*See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one

% instance to run (singleton)".

% See also: GUIDE, GUIDATA, GUIHANDLES

% Begin initialization code - NOT TO BE EDITED

gui\_State =

create struct {'gui\_Name', mfilename, ...

'gui\_Singleton', gui\_Singleton, ...

'gui\_OpeningFcn', @simple\_gui\_OpeningFcn, ...

'gui\_OutputFcn', @simple\_gui\_OutputFcn, ...

'gui\_LayoutFcn', [] , ...

'gui\_Callback', []};

% End initialization code - NOT TO BE EDITED

% --- Executes just before simple\_gui is made visible.

function simple\_gui\_OpeningFcn(hObject, eventdata, handles, varargin)

% This function has no output args, see OutputFcn.

% hObject handle to figure

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% varargin command line arguments to simple\_gui (see VARARGIN)

%initial plotting\_New Algorithm

Start

Open dialog box to read image

Read image

Change to grayscale

Original image = grayscale image

Sharpen the image

Blur the image to get noiseless image

Erode (blurred image); //structural element radius higher than dilation

Dilate (eroded image); //structural element radius lower than erosion

Estimate background

Negate background

Sub (Negative of background from Eroded image)

Sub (Background from Eroded image)

Contrast (Sub (Negative of background from Eroded image))

Contrast (Sub (Background from Eroded image))

Dilate (Contrast (Sub (Background from Eroded image)))

Sub ((Dilate (Contrast (Sub (Background from Eroded image)))) from (Contrast (Sub (Negative of background from Eroded image))))

Tumor = Resulting image

Threshold (Tumor)

Tumor\_Boundary = Boundary (Threshold (Tumor))

Current Data = Original Data

Show(Original Image);

% Choose default command line output for simple\_gui

% Update handles structure

% UIWAIT makes simple\_gui wait for user response (see UIRESUME)

% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.

function varargout = simple\_gui\_OutputFcn(hObject, eventdata, handles)

% varargout cell array for returning output args (see VARARGOUT);

% hObject handle to figure

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure

% --- Executes on selection change in disp\_type.

function disp\_type\_Callback(hObject, eventdata, handles)

% hObject handle to disp\_type (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns disp\_type contents as cell array

% contents{get(hObject,'Value')} returns selected item from disp\_type

val=get(value from drop down list)

Switch (val)

Case 1

Current Data = Original Image

Case 2

Current Data = Tumor

Case 3

Current Data=Tumor\_Boundary;

end

% Update handles structure

% --- Executes during object creation, after setting all properties.

function disp\_type\_CreateFcn(hObject, eventdata, handles)

% hObject handle to disp\_type (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

% --- Executes on button press in clabel.

function clabel\_Callback(hObject, eventdata, handles)

% hObject handle to clabel (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

clabel(contour(Current Data));

% --- Executes on button press in contour.

function contour\_Callback(hObject, eventdata, handles)

% hObject handle to contour (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

Contour (Current Data);

% --- Executes on button press in contourf.

function contourf\_Callback(hObject, eventdata, handles)

% hObject handle to contourf (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

Contour (Current Data);

% --- Executes on button press in new\_img.

function new\_img\_Callback(hObject, eventdata, handles)

% hObject handle to new\_img (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

simple\_gui\_OpeningFcn(~,~, handles);