Automatic hippocampus segmentation of 7.0 Tesla MR images by combining multiple atlases and auto-context models

In many neuroscience and clinical studies, accurate measurement of hippocampus is very important to reveal the inter-subject anatomical differences or the subtle intra-subject longitudinal changes due to aging or dementia. Although many automatic segmentation methods have been developed, their performances are still challenged by the poor image contrast of hippocampus in the MR images acquired especially from 1.5 or 3.0 Tesla (T) scanners. With the recent advance of imaging technology, 7.0 T scanner provides much higher image contrast and resolution for hippocampus study. However, the previousmethods developed for segmenta-tion of hippocampus from 1.5 T or 3.0 T images do not work for the 7.0 T images, due to different levels of im-aging contrast and texture information. In this paper, we present a learning-based algorithm for automatic segmentation of hippocampi from 7.0 T images, by taking advantages of the state-of-the-art multi-atlas frame-work and also the auto-context model (ACM). Specifically, ACM is performed in each atlas domain to iteratively construct sequences of location-adaptive classifiers by integrating both image appearance and local context fea-tures. Due to the plenty texture information in 7.0 T images, more advanced texture features are also extracted and incorporated into the ACM during the training stage. Then, under the multi-atlas segmentation framework, multiple sequences of ACM-based classifiers are trained for all atlases to incorporate the anatomical variability. In the application stage, for a new image, its hippocampus segmentation can be achieved by fusing the labeling re-sults from all atlases, eachofwhich is obtainedbyapplying the atlas-specificACM-based classifiers.

**T**

he segmentation of subfields within the hippocampal formation on in vivo MRI is of major interest because these small anatomic subregions are potentially differentially affected in neu-

ropsychiatric and neurologic disorders, including Alzheimer disease, major depressive disorder, posttraumatic stress disorder, and schizophrenia.1 In the previous decade, 20 segmentation protocols for MRI have been published for the hippocampal subfields and adjacent medial temporal lobe structures.2 Most of these protocols rely on manual segmentation,3-9 which is labor-intensive, requires a long training period, and is often difficult to reproduce between research centers. Automated segmentation methods can help overcome these problems. To our knowledge, currently, only 4 automated segmentation methods exist,10-12 3 of which were developed and evaluated on scans acquired at 3T MR imaging. Only the new FreeSurfer

Please address correspondence to Laura E.M. Wisse, PhD, 3700 Hamilton Walk, Richards Building, Philadelphia, PA 19104; e-mail: Laura.Wisse@uphs.upenn.edu

Indicates open access to non-subscribers at www.ajnr.org Indicates article with supplemental on-line appendix and table.

Indicates article with supplemental on-line photos.

http://dx.doi.org/10.3174/ajnr.A4659

method (http://surfer.nmr.mgh.harvard.edu), developed by Iglesias et al,13 was developed by using a higher resolution 7T postmortem atlas set, though its application has only been demonstrated at lower field strengths. The advantage of in vivo 7T MRI is that high-resolution 3D images can be generated with a relatively short scanning time, making it possible to visualize hippocampal anatomy in greater detail.

Recently, an increasing number of 7T studies have been published on the hippocampal subregional morphology.14-16 Several manual segmentation protocols exist for 7T MRI,5,7,17 and a semi automatic technique for measuring the thickness of hippocampal subfields and layers in the hippocampal body was developed by Kerchner et al.18 In this study, we evaluated the performance of a fully automated segmentation technique for labeling hippocampal subfields and the entorhinal cortex (ERC) at 7T MR imaging, which comes with a new set of challenges, including field inhomogeneity artifacts and increased image size. We do so by adapting a technique previously developed for 3T MRI12 to 7T MRI, labeled by using the manual annotation protocol developed by Wisse et al (2012).5 This protocol and the resulting automatic segmentation cover most of the longitudinal axis of the hippocampal formation. In addition, this article is the first to show that automatic segmentation performs competitively with interrater manual segmentation when the whole length of the hippocampus is labeled. Previously, only Yushkevich et al19 performed a comparison of automatic hippocampal subfield segmentation and interrater manual segmentation reliability, doing so at 3T and only in the body of the hippocampus.

## MATERIALS AND METHODS

### Participants

Participants were included from the PREDICT-MR,16 an ancillary study to the PREDICT-NL study,20 which aimed to investigate determinants and consequences of brain changes on MR imaging in general practice attendees. The cohort included individuals 18 years of age or older who were asked to participate while in the waiting room of their general practitioner, irrespective of their symptoms.

The studies were performed in accordance with the principles of the Declaration of Helsinki and approved by the local ethics committee from the University Medical Center in Utrecht. Written informed consent was obtained from all participants.

### Study Sample for the Atlas Set, Intrarater Reliability, and the Interrater Reliability Set

For the atlas set, 30 participants with a 7T T2-weighted MRI scan, required for the hippocampal subfield segmentation protocol, were randomly selected from the 47 participants in total. Images of 4 were considered to have relatively poor quality due to excessive subject motion, leaving 26 participants for the current study (mean age, 59 9 years; 46% men; median Mini-Mental State Examination score,21 29; range, 25–30).

As a comparison for the reliability of the automated segmentation, we included overlap and reliability values of a single rater (L.E.M.W., rater 1; intrarater reliability) and of 2 raters (L.E.M.W., rater 1, and A.M.H., rater 2; interrater reliability). The intrarater reliability was established in a previous study,5 and the dataset consisted of the first 14 participants of the PREDICT-MR study (overlap with the atlas set, *n*  7).5 For the interrater reliability, a random set of 14 MRI scans of PREDICT-MR was selected for segmentation (overlap with the atlas set, *n* 12). The reliability analysis wasafteratrainingperiod of rater 2 of approximately 5 months, 1 day a week.

See On-line Fig 1 for a Venn diagram describing the samples.

### Image Acquisition

All scans were performed on a 7T MR imaging scanner (Philips Healthcare, Best, the Netherlands) by using a volume transmit coil and a 16-channel receive coil (Nova Medical, Wilmington, Massachusetts) (participants included in the study later than May 2011 were scanned with a volume-transmit and 32-channel receiveheadcoil[NovaMedical]).The7Tprotocolincluded0.70 0.70 0.70 mm3 3D T2-weighted TSE with a TR of 3158 milliseconds, a nominal TE of 301 milliseconds (with a contrast equivalent to a TE of 58 ms for brain tissue in spin-echo sequences with full refocusing angles), a flip angle of 120°(to partly compensate inhomogeneity in the radiofrequency field), a TSE factor of 182, a matrix size of 356 357 272, the application of 2D sensitivity encoding with acceleration factors of 2.0 2.8 (anterior-posterior right-left),andascandurationof10minutesand15seconds.5 Theimageswereinterpolatedbyzero-fillingduringreconstructiontoanominal spatial resolution of 0.35 0.35 0.35 mm3. Moreover, the 7T MRI protocolincludeda1.001.001.00mm3T1-weightedsequencewith aTRof4.8ms,TEof2.2ms,TIof1240ms,aTRoftheinversionpulses of 3500 ms, a matrix size of 200 250 200, and a scan duration of 1 minuteand57seconds.

### Manual Segmentation

The cornu ammonis (CA) fields CA1, CA2, CA3 and the dentate gyrus(DG)(thedentategyruslabelincludesboththegranularcell layer of the dentate gyrus and the hilar region, sometimes called CA4), subiculum (SUB), and ERC were manually segmented, blinded to participant information, by using in-house-developed software22 based on MeVisLab (MeVis Medical Solutions, Bremen, Germany23). Segmentations were performed on coronal images,angulatedperpendiculartothelongaxisofthehippocampal formation. The ERC was segmented according to the protocol by Goncharova et al,24 except for the posterior border, for which we followed the protocol of Insausti et al.25 CA1, CA2, CA3, DG, and SUB were segmented according to a previously published protocol,5 covering most of the long axis of the hippocampal formation. The anterior border was the most anterior section on which the hippocampus could be observed. The posterior border was defined as the section in which the total length of the fornix was visible. This was the most posterior section on which hippocampal subfields were segmented. Beyond this point, subfields fused together and could not be delineated reliably.

### Automated Segmentation

|  |
| --- |
| **FIG 1.** Training and segmentation pipelines in ASHS. Reprinted with permission from Yushkevich et al. 12 Copyright 2014 Wiley Periodicals. |

We applied the automated segmentation of hippocampal subfields (ASHS) technique by using this atlas set. Briefly, the method applies deformable registration of the T1- and T2weighted images,26 multi-atlas joint label fusion,27 and voxelwise learning-based error correction,28 to propagate anatomic labels from a set of manually labeled training images to an unlabeled image. ASHS was evaluated by using a leave-one-out cross-validation (ie, when automatically segmenting the 7T scan of 1 participant in the study, the scans of the remaining 25 participants were used as training data). The resulting automatic segmentation was then compared with the manual segmentation of the same participant. Certain parameters of the method were modified for the 7T segmentation to account for differences in image size and resolution. More details are provided in Fig 1 and the On-line Appendix.

I. INTRODUCTION

Digital image processing is the use of computer algorithms to perform image processing on digital images. Image segmentation is an important and challenging process of image processing. Image segmentation technique is used to partition an image into meaningful parts having similar features and properties. The main aim of segmentation is simplification i.e. representing an image into meaningful and easily analyzable way. Image segmentation is necessary first step in image analysis. The goal of image segmentation is to divide an image into several parts/segments having similar features or attributes. The basic applications of image segmentation are: Content-based image retrieval, Medical imaging, Object detection and Recognition Tasks, Automatic traffic control systems and Video surveillance, etc. The image segmentation can be classified into two basic types: Local segmentation (concerned with specific part or region of image) and Global segmentation (concerned with segmenting the whole image, consisting of large number of pixels). The image segmentation approaches can be categorized into two types based on properties of image.

A. Discontinuity detection based approach

This is the approach in which an image is segmented into regions based on discontinuity. The edge detection based segmentation falls in this category in which edges formed due to intensity discontinuity are detected and linked to form boundaries of regions [1].

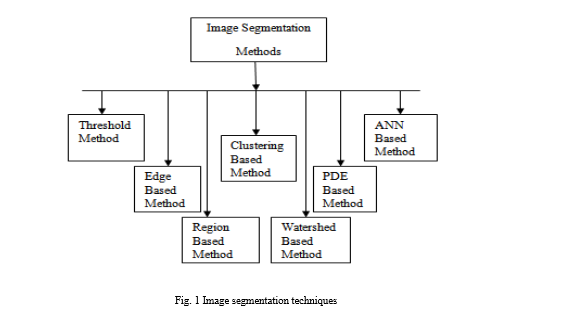
B. Similarity detection based approach

This is the approach in which an image is segmented into regions based on similarity. The techniques that falls under this approach are: thresholding techniques, region growing techniques and region splitting and merging. These all divide the image into regions having similar set of pixels. The clustering techniques also use this methodology. These divide the image into set of clusters having similar features based on some predefined criteria [1] [2].

In other words, also we can say that image segmentation can be approached from three perspectives: Region approach, Edge approach and Data clustering. The region approach falls under similarity detection and edge detection and boundary detection falls under discontinuity detection. Clustering techniques are also under similarity detection.

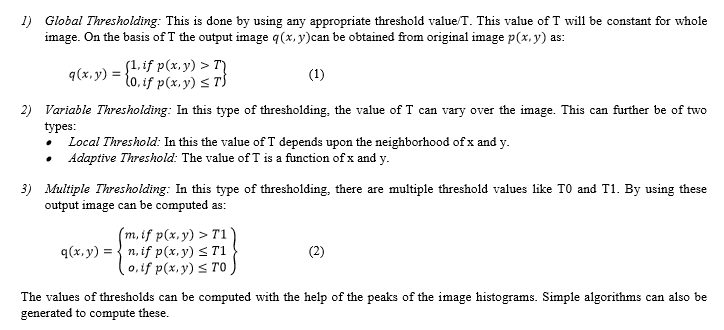
II. CLASSIFICATION OF IMAGE SEGMENTATION TECHNIQUES

There are several existing techniques which are used for image segmentation. These all techniques have their own importance. These all techniques can be approached from two basic approaches of segmentation i.e. region based or edge based approaches. Every technique can be applied on different images to perform required segmentation. These all techniques also can be classified into three categories [3] [4] A. Structural Segmentation Techniques The structural techniques are those techniques of image segmentation that relies upon the information of the structure of required portion of the image i.e. the required region which is to be segmented. B. Stochastic Segmentation Techniques The stochastic techniques are those techniques of the image segmentation that works on the discrete pixel values of the image instead of the structural information of region. C. Hybrid Techniques The hybrid techniques are those techniques of the image segmentation that uses the concepts of both above techniques i.e. these uses discrete pixel and structural information together [5]. In further parts of this paper the various techniques of segmentation are discussed and compared. Mathematical description is avoided for simplicity therefore all the techniques are described theoretically. The popular techniques used for image segmentation are: thresholding method, edge detection based techniques, region based techniques, clustering based techniques, watershed based techniques, partial differential equation based and artificial neural network based techniques etc. These all techniques are different from each other with respect to the method used by these for segmentation



A. Thresholding Method

Thresholding methods are the simplest methods for image segmentation. These methods divide the image pixels with respect to their intensity level. These methods are used over images having lighter objects than background. The selection of these methods can be manual or automatic i.e. can be based on prior knowledge or information of image features. There are basically three types of thresholding [16] [20]:



B. Edge Based Segmentation Method

The edge detection techniques are well developed techniques of image processing on their own. The edge based segmentation methods are based on the rapid change of intensity value in an image because a single intensity value does not provide good information about edges. Edge detection techniques locate the edges where either the first derivative of intensity is greater than a particular threshold or the second derivative has zero crossings. In edge based segmentation methods, first of all the edges are detected and then are connected together to form the object boundaries to segment the required regions. The basic two edge based segmentation methods are: Gray histograms and Gradient based methods. To detect the edges one of the basic edge detection techniques like sobel operator, canny operator and Robert‟s operator etc can be used. Result of these methods is basically a binary image. These are the structural techniques based on discontinuity detection [11].

C. Region Based Segmentation Method

The region based segmentation methods are the methods that segments the image into various regions having similar characteristics. There are two basic techniques based on this method [3] [8] [26].

1) Region growing methods: The region growing based segmentation methods are the methods that segments the image into various regions based on the growing of seeds (initial pixels). These seeds can be selected manually (based on prior knowledge) or automatically (based on particular application). Then the growing of seeds is controlled by connectivity between pixels and with the help of the prior knowledge of problem, this can be stopped. The basic algorithm (based on 8connectivity) steps for region growing method are:

If is the original image that is to be segmented and is the binary image where the seeds are located. Let „T‟ be any predicate which is to be tested for each location.

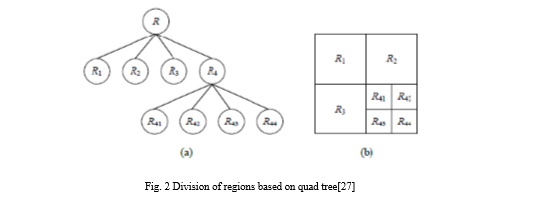
 First of all, all the connected components of „s‟ are eroded.  Compute a binary image PT. Where PT (x, y) = 1, if T(x, y) = True.  Compute a binary image „q‟, where q(x, y) = 1, if PT(x, y) = 1 and (x, y) is 8-connected to seed in „s‟.

These connected components in „q‟ are segmented regions.

2) Region splitting and merging methods: The region splitting and merging based segmentation methods uses two basic techniques i.e. splitting and merging for segmenting an image into various regions. Splitting stands for iteratively dividing an image into regions having similar characteristics and merging contributes to combining the adjacent similar regions. Following diagram shows the division based on quad tree. The basic algorithm steps for region growing and merging are [22].

Let „p‟ be the original image and „T‟ be the particular predicate.

 First of all the R1 is equal to p.  Each region is divided into quadrants for which T (Ri) = False.  If for every region, T (Rj) = True, then merge adjacent regions Ri and Rj such that T (Ri U Rj) = True.  Repeat step 3 until merging is impossible.



D. Clustering Based Segmentation Method

The clustering based techniques are the techniques, which segment the image into clusters having pixels with similar characteristics. Data clustering is the method that divides the data elements into clusters such that elements in same cluster are more similar to each other than others. There are two basic categories of clustering methods: Hierarchical method and Partition based method. The hierarchical methods are based on the concept of trees. In this the root of the tree represents the whole database and the internal nodes represent the clusters. On the other side the partition based methods use optimization methods iteratively to minimize an objective function. In between these two methods there are various algorithms to find clusters. There are basic two types of clustering [13] [24].

1) Hard Clustering: Hard clustering is a simple clustering technique that divides the image into set of clusters such that one pixel can only belong to only one cluster. In other words it can be said that each pixel can belong to exactly one cluster. These methods use membership functions having values either 1 or 0 i.e. one either certain pixel can belong to particular cluster or not. An example of a hard clustering based technique is one k-means clustering based technique known as HCM. In this technique, first of all the centers are computed then each pixel is assigned to nearest center. It emphasizes on maximizing the intra cluster similarity and also minimizing the inter cluster equality.

2) Soft clustering: The soft clustering is more natural type of clustering because in real life exact division is not possible due to the presence of noise. Thus soft clustering techniques are most useful for image segmentation in which division is not strict. The example of such type of technique is fuzzy c-means clustering. In this technique pixels are partitioned into clusters based on partial membership i.e. one pixel can belong to more than one clusters and this degree of belonging is described by membership values. This technique is more flexible than other techniques [13].

E. Watershed Based Methods

The watershed based methods uses the concept of topological interpretation. In this the intensity represents the basins having hole in its minima from where the water spills. When water reaches the border of basin the adjacent basins are merged together. To maintain separation between basins dams are required and are the borders of region of segmentation. These dams areconstructed using dilation. The watershed methods consider the gradient of image as topographic surface. The pixels having more gradient are represented as boundaries which are continuous [15].

F. Partial Differential Equation Based Segmentation Method

The partial differential equation based methods are the fast methods of segmentation. These are appropriate for time critical applications. There are basic two PDE methods: non-linear isotropic diffusion filter (used to enhance the edges) and convex non-quadratic variation restoration (used to remove noise). The results of the PDE method is blurred edges and boundaries that can be shifted by using close operators. The fourth order PDE method is used to reduce the noise from image and the second order PDE method is used to better detect the edges and boundaries [13].

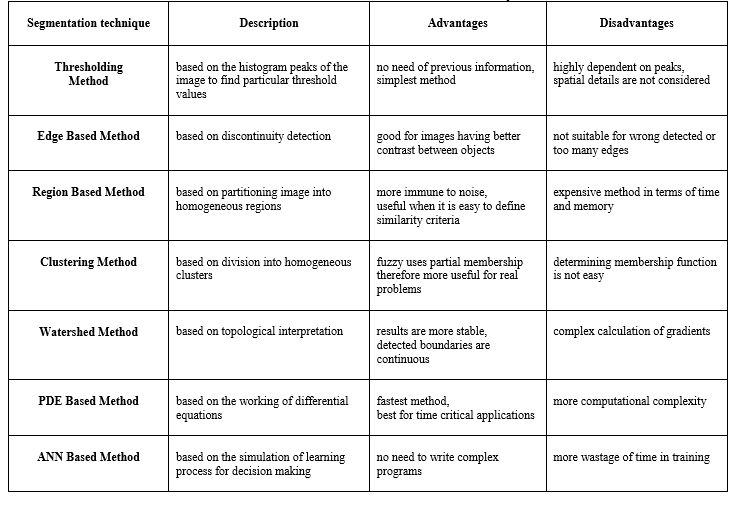
G. Artificial Neural Network Based Segmentation Method

The artificial neural network based segmentation methods simulate the learning strategies of human brain for the purpose of decision making. Now days this method is mostly used for the segmentation of medical images. It is used to separate the required image from background. A neural network is made of large number of connected nodes and each connection has a particular weight. This method is independent of PDE. In this the problem is converted to issues which are solved using neural network. This method has basic two steps: extracting features and segmentation by neural network [8].

COMPARISON

Table I shows a comparison between various segmentation techniques by specifying a brief description of every method each with its advantages and disadvantages[27].

TABLE I COMPARISON OF VARIOUS SEGMENTATION TECHNIQUES



**DIGITAL IMAGE PROCESSING**

**BACKGROUND:**

Digital image processing is an area characterized by the need for extensive experimental work to establish the viability of proposed solutions to a given problem. An important characteristic underlying the design of image processing systems is the significant level of testing & experimentation that normally is required before arriving at an acceptable solution. This characteristic implies that the ability to formulate approaches &quickly prototype candidate solutions generally plays a major role in reducing the cost & time required to arrive at a viable system implementation.

**What is DIP?**

An image may be defined as a two-dimensional function f(x, y), where x & y are spatial coordinates, & the amplitude of f at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y & the amplitude values of f are all finite discrete quantities, we call the image a digital image. The field of DIP refers to processing digital image

by means of digital computer. Digital image is composed of a finite number of elements, each of which has a particular location & value. The elements are called pixels.

Vision is the most advanced of our sensor, so it is not surprising that image play the single most important role in human perception. However, unlike humans, who are limited to the visual band of the EM spectrum imaging machines cover almost the entire EM spectrum, ranging from gamma to radio waves. They can operate also on images generated by sources that humans are not accustomed to associating with image.

There is no general agreement among authors regarding where image processing stops & other related areas such as image analysis& computer vision start. Sometimes a distinction is made by defining image processing as a discipline in which both the input & output at a process are images. This is limiting & somewhat artificial boundary. The area of image analysis (image understanding) is in between image processing & computer vision.

There are no clear-cut boundaries in the continuum from image processing at one end to complete vision at the other. However, one useful paradigm is to consider three types of computerized processes in this continuum: low-, mid-, & high-level processes. Low-level process involves primitive operations such as image processing to reduce noise, contrast enhancement & image sharpening. A low- level process is characterized by the fact that both its inputs & outputs are images. Mid-level process on images involves tasks such as segmentation, description of that object to reduce them to a form suitable for computer processing & classification of individual objects. A mid-level process is characterized by the fact that its inputs generally are images but its outputs are attributes extracted from those images. Finally higher- level processing involves “Making sense” of an ensemble of recognized objects, as in image analysis & at the far end of the continuum performing the cognitive functions normally associated with human vision.

Digital image processing, as already defined is used successfully in a broad range of areas of exceptional social & economic value.

**What is an image?**

An image is represented as a two dimensional function f(x, y) where x and y are spatial co-ordinates and the amplitude of ‘f’ at any pair of coordinates (x, y) is called the intensity of the image at that point.

**Gray scale image:**

A grayscale image is a function I(xylem) of the two spatial coordinates of the image plane.

I(x, y) is the intensity of the image at the point (x, y) on the image plane.

I (xylem)takes non-negative values assume the image is bounded by arectangle[0, a] ×[0, b]I: [0, a] × [0, b] → [0, info)

**Color image:**

It can be represented by three functions, R (xylem)for red,G (xylem)for green *and* B (xylem)for blue.

An image may be continuous with respect to the x and y coordinates and also in amplitude. Converting such an image to digital form requires that the coordinates as well as the amplitude to be digitized. Digitizing the coordinate’s values is called sampling. Digitizing the amplitude values is called quantization.

**Coordinate convention:**

The result of sampling and quantization is a matrix of real numbers. We use two principal ways to represent digital images. Assume that an image f(x, y) is sampled so that the resulting image has M rows and N columns. We say that the image is of size M X N. The values of the coordinates (xylem) are discrete quantities. For notational clarity and convenience, we use integer values for these discrete coordinates. In many image processing books, the image origin is defined to be at (xylem)=(0,0).The next coordinate values along the first row of the image are (xylem)=(0,1).It is important to keep in mind that the notation (0,1) is used to signify the second sample along the first row. It does not mean that these are the actual values of physical coordinates when the image was sampled. Following figure shows the coordinate convention. Note that x ranges from 0 to M-1 and y from 0 to N-1 in integer increments.

The coordinate convention used in the toolbox to denote arrays is different from the preceding paragraph in two minor ways. First, instead of using (xylem) the toolbox uses the notation (race) to indicate rows and columns. Note, however, that the order of coordinates is the same as the order discussed in the previous paragraph, in the sense that the first element of a coordinate topples, (alb), refers to a row and the second to a column. The other difference is that the origin of the coordinate system is at (r, c) = (1, 1); thus, r ranges from 1 to M and c from 1 to N in integer increments. IPT documentation refers to the coordinates. Less frequently the toolbox also employs another coordinate convention called spatial coordinates which uses x to refer to columns and y to refers to rows. This is the opposite of our use of variables x and y.

**Image as Matrices:**

The preceding discussion leads to the following representation for a digitized image function:

f (0,0) f(0,1) ……….. f(0,N-1)

f(1,0) f(1,1) ………… f(1,N-1)

f(xylem)= . . .

. . .

f(M-1,0) f(M-1,1) ………… f(M-1,N-1)

The right side of this equation is a digital image by definition. Each element of this array is called an image element, picture element, pixel or pel. The terms image and pixel are used throughout the rest of our discussions to denote a digital image and its elements.

A digital image can be represented naturally as a MATLAB matrix:

f(1,1) f(1,2) ……. f(1,N)

f(2,1) f(2,2) …….. f(2,N) . . .

f = f(M,1) f(M,2) …….f(M,N)

Where f(1,1) = f(0,0) (note the use of a monoscope font to denote MATLAB quantities). Clearly the two representations are identical, except for the shift in origin. The notation f(p ,q) denotes the element located in row p and the column q. For example f(6,2) is the element in the sixth row and second column of the matrix f. Typically we use the letters M and N respectively to denote the number of rows and columns in a matrix. A 1xN matrix is called a row vector whereas an Mx1 matrix is called a column vector. A 1x1 matrix is a scalar.

Matrices in MATLAB are stored in variables with names such as A, a, RGB, real array and so on. Variables must begin with a letter and contain only letters, numerals and underscores. As noted in the previous paragraph, all MATLAB quantities are written using mono-scope characters. We use conventional Roman, italic notation such as f(x ,y), for mathematical expressions

**Reading Images:**

Images are read into the MATLAB environment using function imread whose syntax is

imread(‘filename’)

Format name Description recognized extension

TIFF Tagged Image File Format .tif, .ti

JPEG Joint Photograph Experts Group .jpg, .jpeg

GIF Graphics Interchange Format .gif

BMP Windows Bitmap .bmp

PNG Portable Network Graphics .png

XWD X Window Dump .xwd

Here filename is a spring containing the complete of the image file(including any applicable extension).For example the command line

>> f = imread (‘8. jpg’);

reads the JPEG (above table) image chestxray into image array f. Note the use of single quotes (‘) to delimit the string filename. The semicolon at the end of a command line is used by MATLAB for suppressing output. If a semicolon is not included. MATLAB displays the results of the operation(s) specified in that line. The prompt symbol(>>) designates the beginning of a command line, as it appears in the MATLAB command window.

When as in the preceding command line no path is included in filename, imread reads the file from the current directory and if that fails it tries to find the file in the MATLAB search path. The simplest way to read an image from a specified directory is to include a full or relative path to that directory in filename.

For example,

>> f = imread ( ‘D:\myimages\chestxray.jpg’);

reads the image from a folder called my images on the D: drive, whereas

>> f = imread(‘ . \ myimages\chestxray .jpg’);

reads the image from the my images subdirectory of the current of the current working directory. The current directory window on the MATLAB desktop toolbar displays MATLAB’s current working directory and provides a simple, manual way to change it. Above table lists some of the most of the popular image/graphics formats supported by imread and imwrite.

**Data Classes:**

Although we work with integers coordinates the values of pixels themselves are not restricted to be integers in MATLAB. Table above list various data classes supported by MATLAB and IPT are representing pixels values. The first eight entries in the table are refers to as numeric data classes. The ninth entry is the char class and, as shown, the last entry is referred to as logical data class.

All numeric computations in MATLAB are done in double quantities, so this is also a frequent data class encounter in image processing applications. Class unit 8 also is encountered frequently, especially when reading data from storages devices, as 8 bit images are most common representations found in practice. These two data classes, classes logical, and, to a lesser degree, class unit 16 constitute the primary data classes on which we focus. Many ipt functions however support all the data classes listed in table. Data class double requires 8 bytes to represent a number uint8 and int 8 require one byte each, uint16 and int16 requires 2bytes and unit 32.

**Image Types:**

The toolbox supports four types of images:

1 .Intensity images

2. Binary images

3. Indexed images

4. R G B images

Most monochrome image processing operations are carried out using binary or intensity images, so our initial focus is on these two image types. Indexed and RGB colour images.

**Intensity Images:**

An intensity image is a data matrix whose values have been scaled to represent intentions. When the elements of an intensity image are of class unit8, or class unit 16, they have integer values in the range [0,255] and [0, 65535], respectively. If the image is of class double, the values are floating \_point numbers. Values of scaled, double intensity images are in the range [0, 1] by convention.

**Binary Images:**

Binary images have a very specific meaning in MATLAB.A binary image is a logical array 0s and1s.Thus, an array of 0s and 1s whose values are of data class, say unit8, is not considered as a binary image in MATLAB .A numeric array is converted to binary using function logical. Thus, if A is a numeric array consisting of 0s and 1s, we create an array B using the statement.

B=logical (A)

If A contains elements other than 0s and 1s.Use of the logical function converts all nonzero quantities to logical 1s and all entries with value 0 to logical 0s.

Using relational and logical operators also creates logical arrays.

To test if an array is logical we use the I logical function:

islogical(c)

If c is a logical array, this function returns a 1.Otherwise returns a 0. Logical array can be converted to numeric arrays using the data class conversion functions.

**Indexed Images:**

An indexed image has two components:

A data matrix integer, x.

A color map matrix, map.

Matrix map is an m\*3 arrays of class double containing floating\_ point values in the range [0, 1].The length m of the map are equal to the number of colors it defines. Each row of map specifies the red, green and blue components of a single color. An indexed images uses “direct mapping” of pixel intensity values color map values. The color of each pixel is determined by using the corresponding value the integer matrix x as a pointer in to map. If x is of class double ,then all of its components with values less than or equal to 1 point to the first row in map, all components with value 2 point to the second row and so on. If x is of class units or unit 16, then all components value 0 point to the first row in map, all components with value 1 point to the second and so on.

**RGB Image:**

An RGB color image is an M\*N\*3 array of color pixels where each color pixel is triplet corresponding to the red, green and blue components of an RGB image, at a specific spatial location. An RGB image may be viewed as “stack” of three gray scale images that when fed in to the red, green and blue inputs of a color monitor

Produce a color image on the screen. Convention the three images forming an RGB color image are referred to as the red, green and blue components images. The data class of the components images determines their range of values. If an RGB image is of class double the range of values is [0, 1].

Similarly the range of values is [0,255] or [0, 65535].For RGB images of class units or unit 16 respectively. The number of bits use to represents the pixel values of the component images determines the bit depth of an RGB image. For example, if each component image is an 8bit image, the corresponding RGB image is said to be 24 bits deep.

Generally, the number of bits in all component images is the same. In this case the number of possible color in an RGB image is (2^b) ^3, where b is a number of bits in each component image. For the 8bit case the number is 16,777,216 colors

**INTRODUCTION TO MATLAB**

**What Is MATLAB?**

MATLAB® is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include

Math and computation

Algorithm development

Data acquisition

Modeling, simulation, and prototyping

Data analysis, exploration, and visualization

Scientific and engineering graphics

Application development, including graphical user interface building.

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or FORTRAN.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most users of MATLAB, toolboxes allow you to learnand apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

**The MATLAB System:**

The MATLAB system consists of five main parts:

**Development Environment:**

 This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and Command Window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

**The MATLAB Mathematical Function:**

This is a vast collection of computational algorithms ranging from elementary functions like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix eigen values, Bessel functions, and fast Fourier transforms.

**The MATLAB Language:**

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create complete large and complex application programs.

**Graphics:**

MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

**The MATLAB Application Program Interface (API):**

This is a library that allows you to write C and Fortran programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

**MATLAB working environment:**

**MATLAB desktoP:-**

Matlab Desktop is the main Matlab application window. The desktop contains five sub windows, the command window, the workspace browser, the current directory window, the command history window, and one or more figure windows, which are shown only when the user displays a graphic.

The command window is where the user types MATLAB commands and expressions at the prompt (>>) and where the output of those commands is displayed. MATLAB defines the workspace as the set of variables that the user creates in a work session. The workspace browser shows these variables and some information about them. Double clicking on a variable in the workspace browser launches the Array Editor, which can be used to obtain information and income instances edit certain properties of the variable.

The current Directory tab above the workspace tab shows the contents of the current directory, whose path is shown in the current directory window. For example, in the windows operating system the path might be as follows: C:\MATLAB\Work, indicating that directory “work” is a subdirectory of the main directory “MATLAB”; WHICH IS INSTALLED IN DRIVE C. clicking on the arrow in the current directory window shows a list of recently used paths. Clicking on the button to the right of the window allows the user to change the current directory.

MATLAB uses a search path to find M-files and other MATLAB related files, which are organize in directories in the computer file system. Any file run in MATLAB must reside in the current directory or in a directory that is on search path. By default, the files supplied with MATLAB and math works toolboxes are included in the search path. The easiest way to see which directories are on the search path. The easiest way to see which directories are soon the search paths, or to add or modify a search path, is to select set path from the File menu the desktop, and then use the set path dialog box. It is good practice to add any commonly used directories to the search path to avoid repeatedly having the change the current directory.

The Command History Window contains a record of the commands a user has entered in the command window, including both current and previous MATLAB sessions. Previously entered MATLAB commands can be selected and re-executed from the command history window by right clicking on a command or sequence of commands.

This action launches a menu from which to select various options in addition to executing the commands. This is useful to select various options in addition to executing the commands. This is a useful feature when experimenting with various commands in a work session.

**Using the MATLAB Editor to create M-Files:**

The MATLAB editor is both a text editor specialized for creating M-files and a graphical MATLAB debugger. The editor can appear in a window by itself, or it can be a sub window in the desktop. M-files are denoted by the extension .m, as in pixelup.m. The MATLAB editor window has numerous pull-down menus for tasks such as saving, viewing, and debugging files. Because it performs some simple checks and also uses color to differentiate between various elements of code, this text editor is recommended as the tool of choice for writing and editing M-functions. To open the editor, type edit at the prompt opens the M-file filename.m in an editor window, ready for editing. As noted earlier, the file must be in the current directory, or in a directory in the search path.

Image Representation 3.1 Image Format An image is a rectangular array of values (pixels). Each pixel represents the measurement of some property of a scene measured over a finite area. The property could be many things, but we usually measure either the average brightness (one value) or the brightnesses of the image filtered through red, green and blue filters (three values). The values are normally represented by an eight bit integer, giving a range of 256 levels of brightness. We talk about the resolution of an image: this is defined by the number of pixels and number of brightness values. A raw image will take up a lot of storage space. Methods have been defined to compress the image by coding redundant data in a more efficient fashion, or by discarding the perceptually less significant information. MATLAB supports reading all of the common image formats. Image coding is not addressed in this course unit.

3.2 Image Loading and Displaying and Saving An image is loaded into working memory using the command

>> f = imread(‘image file name’);

The semicolon at the end of the command suppresses MATLAB output. Without it, MATLAB will execute the command and echo the results to the screen. We assign the image to the array f. If no path is specified, MATLAB will look for the image file in the current directory. The image can be displayed using >> imshow(f, G)

f is the image to be displayed, G defines the range of intensity levels used to display it. If it is omitted, the default value 256 is used. If the syntax [low, high] is used instead of G, values less than low are displayed as black, and ones greater than high are displayed as white. Finally, if low and high are left out, i.e. use [ ], low is set to the minimum value in the image and high to the maximum one, which is useful for automatically fixing the range of the image if it is very small or vary large. Images are usually displayed in a figure window. If a second image is displayed it will overwrite the first, unless the figure function is used:

>> figure, imshow(f)

will generate a new figure window and display the image in it. Note that multiple functions may be called in a single line, provided they are separated by commas. An image array may be written to file using:

>> imwrite(array name, ‘file name’)

The format of the file can be inferred from the file extension, or can be specified by a third argument. Certain file formats have additional arguments. 3.3 Image Information Information about an image file may be found by

>> imfinfo filename

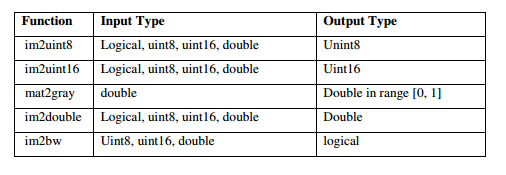
4 Quantisation 4.1 Grey Level Ranges Images are normally captured with pixels in each channel being represented by eight bit integers. (This is partly for historical reasons – it has the convenience of being a basic memory unit, it allows for a suitable range of values to be represented, and many cameras could not capture data to any greater accuracy. Further, most displays are limited to eight bits per red, green and blue channel.) But there is no reason why pixels should be so limited, indeed, there are devices and applications that deliver and require higher resolution data.

illumination (other wavelength ranges and more of them are possible). MATLAB provides functions for changing images from one type to another. The syntax is

>> B = data\_class\_name(A)

where data\_class\_name is one of the data types in the above table, e.g.

>> B = uint8(A)

**

4.2 Number of Pixels Images come in all sizes, but are (almost) always rectangular. MATLAB gives several methods of accessing the elements of an array, i.e. the pixels of an image. An element can be accessed directly: typing the array name at the prompt will return all the array elements (which could take a while), typing the array name followed by element indices in round brackets, will return that value. Ranges of array elements can be accessed using colons.

Ranges of array elements can be accessed using colons.

>> A(first:last)

Will return the first to last elements inclusive of the one dimensional array A. Note that the indices start at one.

>> A(first : step : last)

Will return every step elements starting from first and finishing when last is reached or exceeded. Step could be negative, in which case you’d have to ensure that first was greater than last. Naturally, this notation can be extended to access portions of an image. An image, f, could be flipped using

>> fp = f(end : -1 : 1, :);

The keyword end is used to signify the last index. Using the colon alone implies that all index values are traversed. This also indicates how multi-dimensional arrays are accessed. Or a section of an image could be abstracted using

>> fc = f(top : bottom, left : right);

Or the image could be subsampled using

>> fs = f(1 : 2 : end, 1 : 2 : end);

4.2.1 A note on colour images. If the input image is colour, these operations will return greyscale results. A colour image has three values per pixel, which are accessed using a third index.

>> A(x, y, 1:3)

Would return all three colour values of the pixel at (x,y). A colour plane could be abstracted using

>> R = A(x, y, 1);

And similarly for G (last index = 2) and B.

Point Processing Point processing operations manipulate individual pixel values, without regard to any neighbouring values. Two types of transforms can be identified, manipulating the two properties of a pixel: its value and position.

5.1 Value Manipulation The fundamental value of a pixel is its brightness (in a monochrome image) or colour (in a multichannel image).

5.1.1 Pixel Scaling Scaling of pixel values is achieved by multiplying by a constant. MATLAB provides a single function that achieves several effects

>> R = imadjust(A, [low\_in, high\_in], [low\_out, high\_out], gamma);

This takes the input range of values as specified and maps them to the output range that’s specified. Values outside of the input range are clamped to the extremes of the output range (values below low\_in are all mapped to low\_out). The range values are expected to be in the interval [0, 1]. The function scales them to values appropriate to the image type before applying the scaling operation. Whilst low\_in is expected to be less than high\_in, the same is not true for low\_out and high\_out. The image can therefore be inverted. The value of gamma specifies the shape of the mapped curve. Gamma = 1 gives a linear scaling, a smaller gamma gives a mapping that expands the scale at lower values, a larger gamma expands the upper range of the scale. This can make the contrast between darker or brighter tones more visible, respectively. Omitting any of the parameters results in default values being assumed. The extremes of the ranges are used (0 and 1), or gamma = 1.

**Histogram** The histogram of an image measures the number of pixels with a given grey or colour value. Histograms of colour images are not normally used, so will not be discussed here. The histogram of an image with L distinct intensity levels in the range [0, G] is defined as the function h(rk) = nk.

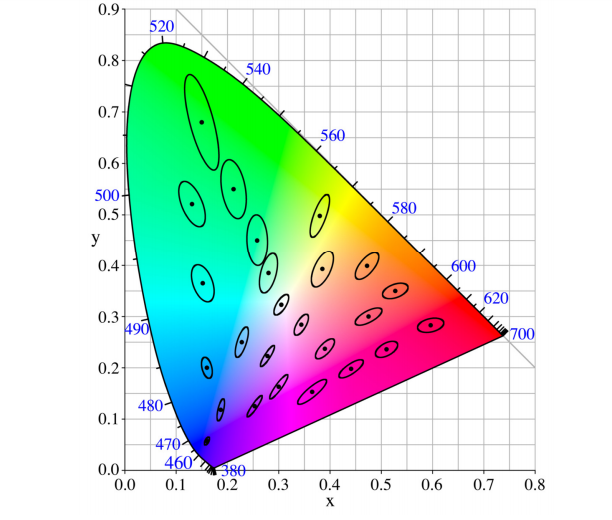
rk is the k th intensity level in the image, and nk will be the number of pixels with grey value rk . G will be 255 for a uint8 image, 65536 for uint16 and 1.0 for a double image. Since the lower index of MATLAB arrays is one, never zero, r1 will correspond to intensity level 0, etc. For the integer valued images, G = L-1. We often work with normalised histograms. A normalised histogram is obtained by dividing each element of h(rk) by the total number of pixels in the image (equal to the sum of histogram elements). Such a histogram is called the probability density function (pdf) and reflects the probability of a given intensity level occurring. p(rk) = nk /n MATLAB functions for computing and displaying the histogram and normalised histogram are: >> h = imhist(A, b); >> p = imhist(A, b)/numel(A); b is the number of bins in the histogram. If it is omitted, 256 is assumed. The function numel(A) returns the number of elements in the argument, in this case the number of pixels in A. Additional functions are defined for displaying the data in different forms: notably as a bar graph and for adding axes.

Histogram Equalisation It is possible to manipulate the grey scale to achieve different effects. One of these effects to known as histogram equalisation. The aim is to transform the grey scale such that the pdf of the output image is uniform. Some authors claim that this improves the appearance of the image. The transformation is achieved using: >> h = histeq(A, nlev); Where nlev is the number of levels in the output image, h. Its default value is 64.

Thresholding This is a simple method of differentiating between an object and the background, which works provided they are of different intensities. A threshold value is defined. Intensities greater than this are set to one value, intensities less than to another (1 or max, and 0 or min are often used). Whilst a threshold can be decided manually, it is better to use the image data to compute one. Gonzalez and Woods suggested this method: 1. Choose a threshold arbitrarily. 2. Threshold the image using it. 3. Compute the mean grey value of the pixels with intensities above and below the threshold, and then compute the average of these two values. 4. Use the new value to rethreshold the image. 5. Repeat steps 3 and 4 until the threshold changes by an insignificant amount. Otsu suggested a method than minimised the between class variance. This is the method provided by the MATLAB toolbox: >> T = graythresh(A);T is a normalised value between 0.0 and 1.0; it should be scaled to the proper range before using it. The conversion function im2bw will then return the thresholded image.

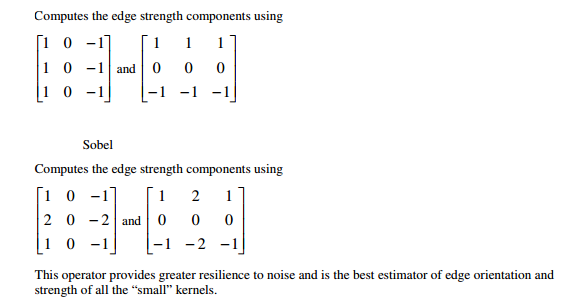
5.1.5 Colour Transforms Multiple methods of representing colour data exist. Whilst RGB is most widely used for capture and display, it is not always the best for image processing, since it is a perceptually non-uniform representation. This means that if we change the RGB values by a fixed amount, the observed difference depends on the original RGB values. One way of observing this is to mix the output of standardised coloured lights to generate a colour, then alter the brightness of the input until an observer just notices a change in the light’s colour. The original colour and the colour of the just noticeable difference can be plotted. By making measurements systematically over the whole colour space, we can generate a MacAdam diagram. The points represent the original colour, the ellipses the just noticeable difference contours. It is also possible to categorise colour spaces as being device dependent or device independent. Device dependent spaces are used in the broadcast and printing industry, largely for convenience. The most widely used spaces are YIQ, YCrCb and HSV. Conversion between the spaces is by using simple functions. E.g. >> YIQ = rgb2ntsc(RGB); Device independent spaces are used because the device dependent spaces include subjective definitions. The CIE defined a standardised colour space in 1931. It specifies three colour sources, called X, Y and Z. All visible colours can be generated by a linear combination of these. The X, Y and Z values can be normalised, to sum to 1. The colours represented by the normalised x and y values can be plotted – as in the MacAdam diagram. Conversion of data between colour spaces is a two stage process. A colour transformation structure is first defined, e.g. to convert from RGB to XYZ: >> C = makecform(‘srgb2xyz’); And then perform the conversion

>> Ixyz = applycform(Irgb, C); The data types used to represent the data may alter. Other colour spaces have been defined, each attempting to make the colour space more perceptually uniform.



5.2 Co-ordinate Manipulation The result of co-ordinate manipulation is to distort an image. For example, in creating panoramic images, or correcting images for lens distortion in order to make measurements. Manipulation has two stages – computing the pixels’ new co-ordinates and resampling, or interpolation. 5.2.1 Transform There are two classes of transform: affine and non-linear. Affine transforms are achieved by inserting suitable values into a transform matrix:

High Pass - Edge Detection An edge is defined as a significant, local change in image intensity. The simplest (ideal) edge is a line (the edge) separating uniform regions of different intensity. In practice, this is never observed as noise superimposes random fluctuations on the image data, and imperfect optics and digitisation blur the image. The ideal edge is therefore corrupted and we see a rather more gradual change in intensity from one region to the other. The problem of edge detection is to locate the edge in this data (if it is possible, and if it makes sense). If an edge is a discontinuity in intensity, there must be a region around it where the intensity changes by large amounts over small distances, i.e. the gradient is high. The gradient can be estimated by measuring the differences in intensity over small distances and dividing by the distance – this is the digital equivalent of differentiation. Rather than take differences between individual pixels, local averages are computed and compared as this is less susceptible to the noise that is present. An edge can take any orientation within an image. Differentiation can only be performed in directions parallel to the two axes. The edge strength and orientation can be estimated as described in Appendix B. Once the convolution has been performed and edge strengths estimated, we obtain a grey scale image whose intensity is proportional to the likelihood of a pixel being on an edge. But if the edge has been smeared out, where is the “true” edge? Non-maximal suppression is used to remove pixels that are not located on the position of maximum slope: we can track across an edge using the edge orientation information and locate the pixel with the greatest edge strength. 6.1.1 Prewitt Computes the edge strength components using



6.1.3 Canny Canny took an information theoretic approach to edge detection, stating that an edge detector should 1. Detect an edge 2. Should give a response in the correct location 3. Have a single response to an edge He assumed that he was searching for an ideal step edge in the presence of Gaussian noise and defined a matched filter that could be approximated by the difference of a Gaussian. Although this gives the optimal edge detector for images with only this type of noise corruption, it responds adequately for other noise distributions. The operator first convolves the image with a Gaussian kernel to perform the noise reduction (just as the Prewitt operator has the regions of 1s to perform averaging). It then differentiates the image in the two orientations. Rather than perform one convolution for smoothing and a further two for differentiation, the smoothing and differentiation kernels are combined and the whole operation is performed using two convolutions. (this works because convolution is associative: ∇ • (G • I) = (∇ •G)• I .) The amount of noise reduction may be controlled by varying the standard deviation, σ, in the Gaussian function. Larger values of σ imply more smoothing: more noise reduction and also more blurring of the edge information.

Output screens:-

## CONCLUSIONS

We present a fully automated segmentation method of hippocampal subfields at 7T MRI with high accuracy for most of the subfields. The accuracy of this method is competitive with other published automated methods and with the interrater reliability formanualsegmentation.

## REFERENCES

1. Small SA, Schobel SA, Buxton RB, et al. **A pathophysiological framework of hippocampal dysfunction in ageing and disease.** *Nat Rev Neurosci* 2011;12:585–601 [CrossRef](http://dx.doi.org/%2010.1038/nrn3085) [Medline](http://www.ncbi.nlm.nih.gov/pubmed/21897434)
2. Yushkevich PA, Amaral RS, Augustinack JC, et al; Hippocampal Subfields Group (HSG). **Quantitative comparison of 21 protocols for labeling hippocampal subfields and parahippocampal subregions ininvivoMRI:towardsaharmonizedsegmentationprotocol.***Neuroimage* 2015;111:526–41 [CrossRef](http://dx.doi.org/10.1016/j.neuroimage.2015.01.004) [Medline](http://www.ncbi.nlm.nih.gov/pubmed/25596463)
3. Mueller SG, Stables L, Du AT, et al. **Measurement of hippocampal subfields and age-related changes with high resolution MRI at 4T.** *Neurobiol Aging* 2007;28:719–26 [CrossRef](http://dx.doi.org/10.1016/j.neurobiolaging.2006.03.007) [Medline](http://www.ncbi.nlm.nih.gov/pubmed/16713659)
4. Malykhin NV, Lebel RM, Coupland NJ, et al. **In vivo quantification ofhippocampalsubfieldsusing4.7Tfastspinechoimaging.***Neuroimage* 2010;49:1224–30 [CrossRef](http://dx.doi.org/10.1016/j.neuroimage.2009.09.042) [Medline](http://www.ncbi.nlm.nih.gov/pubmed/19786104)
5. Wisse LE, Gerritsen L, Zwanenburg JJ, et al. **Subfields of the hippocampal formation at 7T MRI: in vivo volumetric assessment.** *Neuroimage* 2012;61:1043–49 [CrossRef](http://dx.doi.org/10.1016/j.neuroimage.2012.03.023) [Medline](http://www.ncbi.nlm.nih.gov/pubmed/22440643)
6. La Joie R, Fouquet M, Me´zenge F, et al. **Differential effect of age on hippocampalsubfieldsassessedusinganewhigh-resolution3TMR sequence.** *Neuroimage* 2010;53:506–14 [CrossRef](http://dx.doi.org/10.1016/j.neuroimage.2010.06.024) [Medline](http://www.ncbi.nlm.nih.gov/pubmed/20600996)
7. Kerchner GA, Hess CP, Hammond-Rosenbluth KE, et al. **Hippocampal CA1 apical neuropil atrophy in mild Alzheimer disease visualized with 7-T MRI.** *Neurology* 2010;75:1381–87 [CrossRef](http://dx.doi.org/10.1212/WNL.0b013e3181f736a1) [Medline](http://www.ncbi.nlm.nih.gov/pubmed/20938031)
8. Raz N, Daugherty AM, Bender AR, et al. **Volumeofthehippocampal subfields in healthy adults: differential associations with age and a pro-inflammatory genetic variant.** *Brain Struct Funct* 2015;220: 2663–74 [CrossRef](http://dx.doi.org/10.1007/s00429-014-0817-6) [Medline](http://www.ncbi.nlm.nih.gov/pubmed/24947882)
9. Winterburn JL, Pruessner JC, Chavez S, et al. **A novel in vivo atlas of human hippocampal subfields using high-resolution 3 T magnetic resonance imaging.** *Neuroimage* 2013;74:254–65 [CrossRef](http://dx.doi.org/10.1016/j.neuroimage.2013.02.003) [Medline](http://www.ncbi.nlm.nih.gov/pubmed/23415948)
10. Van Leemput K, Bakkour A, Benner T, et al. **Automated segmentation of hippocampal subfields from ultra-high resolution in vivo MRI.** *Hippocampus* 2009;19:549–57 [CrossRef](http://dx.doi.org/10.1002/hipo.20615) [Medline](http://www.ncbi.nlm.nih.gov/pubmed/19405131)
11. Pipitone J, Park MT, Winterburn J, et al; Alzheimer’s Disease Neuroimaging Initiative. **Multi-atlas segmentation of the whole hippocampus and subfields using multiple automatically generated templates.** *Neuroimage* 2014;101:494–512 [CrossRef](http://dx.doi.org/10.1016/j.neuroimage.2014.04.054) [Medline](http://www.ncbi.nlm.nih.gov/pubmed/24784800)
12. Yushkevich PA, Pluta JB, Wang H, et al. **Automated volumetry and regional thickness analysis of hippocampal subfields and medial temporal cortical structures in mild cognitive impairment.** *Hum Brain Mapp* 2015;36:258–87 [CrossRef](http://dx.doi.org/10.1002/hbm.22627) [Medline](http://www.ncbi.nlm.nih.gov/pubmed/25181316)
13. IglesiasJE,AugustinackJC,NguyenK,etal;Alzheimer’sDiseaseNeuroimaging Initiative. **A computational atlas of the hippocampal formation using ex vivo, ultra-high resolution MRI: application to adaptivesegmentationofinvivoMRI.***Neuroimage*2015;115:117–37 [CrossRef](http://dx.doi.org/10.1016/j.neuroimage.2015.04.042) [Medline](http://www.ncbi.nlm.nih.gov/pubmed/25936807)
14. Cho ZH, Han JY, Hwang SI, et al. **Quantitative analysis of the hippocampususingimagesobtainedfrom7.0TMRI.***Neuroimage*2010; 49:2134–40 [CrossRef](http://dx.doi.org/10.1016/j.neuroimage.2009.11.002) [Medline](http://www.ncbi.nlm.nih.gov/pubmed/19909820)
15. Thomas BP, Welch EB, Niederhauser BD, et al. **High-resolution 7T MRI of the human hippocampus in vivo.** *J Magn Reson Imaging* 2008;28:1266–72 [CrossRef](http://dx.doi.org/10.1002/jmri.21576) [Medline](http://www.ncbi.nlm.nih.gov/pubmed/18972336)
16. Wisse LE, Biessels GJ, Heringa SM, et al; Utrecht Vascular CognitiveImpairment (VCI) Study Group. **Hippocampal subfield volumes at 7T in early Alzheimer’s disease and normal aging.** *Neurobiol Aging* 2014;35:2039–45 [CrossRef](http://dx.doi.org/10.1016/j.neurobiolaging.2014.02.021) [Medline](http://www.ncbi.nlm.nih.gov/pubmed/24684788)
17. Boutet C, Chupin M, Lehe´ricy S, et al. **Detection of volume loss in hippocampallayersinAlzheimer’sdiseaseusing7TMRI:afeasibility study.** *Neuroimage Clin* 2014;5:341–48 [CrossRef](http://dx.doi.org/10.1016/j.nicl.2014.07.011) [Medline](http://www.ncbi.nlm.nih.gov/pubmed/25161900)
18. KerchnerGA,DeutschGK,ZeinehM,etal.**HippocampalCA1apical neuropilatrophyandmemoryperformanceinAlzheimer’sdisease.** *Neuroimage* 2012;63:194–202 [CrossRef](http://dx.doi.org/10.1016/j.neuroimage.2012.06.048) [Medline](http://www.ncbi.nlm.nih.gov/pubmed/22766164)
19. Yushkevich PA, Wang H, Pluta J, et al. **Nearly automatic segmentation of hippocampal subfields in in vivo focal T2-weighted MRI.** *Neuroimage* 2010;53:1208–24 [CrossRef](http://dx.doi.org/10.1016/j.neuroimage.2010.06.040) [Medline](http://www.ncbi.nlm.nih.gov/pubmed/20600984)
20. Stegenga BT, Kamphuis MH, King M, et al. **The natural course and outcome of major depressive disorder in primary care: the PREDICT-NL study.** *Soc Psychiatry Psychiatr Epidemiol* 2012;47: 87–95 [CrossRef](http://dx.doi.org/10.1007/s00127-010-0317-9) [Medline](http://www.ncbi.nlm.nih.gov/pubmed/21057769)
21. Folstein MF, Folstein SE, McHugh PR. **“Mini-mental state”: a practicalmethodforgradingthecognitivestateofpatientsfortheclinician.** *J Psychiatr Res* 1975;12:189–98 [CrossRef](http://dx.doi.org/10.1016/0022-3956(75)90026-6) [Medline](http://www.ncbi.nlm.nih.gov/pubmed/1202204)
22. Kuijf HJ. *Image Processing Techniques for Quantification and Assessment of Brain MRI* [dissertation]. Utrecht: Utrecht University Repository; 2013
23. Ritter F, Boskamp T, Homeyer A, et al. **Medical image analysis.** *IEEE Pulse* 2011;2:60–70 [CrossRef](http://dx.doi.org/10.1109/MPUL.2011.942929) [Medline](http://www.ncbi.nlm.nih.gov/pubmed/22147070)
24. Goncharova II, Dickerson BC, Stoub TR, et al. **MRI of human entorhinalcortex:areliableprotocolforvolumetricmeasurement.***Neurobiol Aging* 2001;22:737–45 [CrossRef](http://dx.doi.org/10.1016/S0197-4580(01)00270-6) [Medline](http://www.ncbi.nlm.nih.gov/pubmed/11705633)
25. Insausti R, Juottonen K, Soininen H, et al. **MR volumetric analysis of the human entorhinal, perirhinal, and temporopolar cortices.** *AJNR Am J Neuroradiol* 1998;19:659–71 [Medline](http://www.ncbi.nlm.nih.gov/pubmed/9576651)
26. Avants BB, Epstein CL, Grossman M, et al. **Symmetric diffeomorphic image registration with cross-correlation: evaluating automated labeling of elderly and neurodegenerative brain.** *Med Image Anal* 2008;12:26–41 [CrossRef](http://dx.doi.org/10.1016/j.media.2007.06.004) [Medline](http://www.ncbi.nlm.nih.gov/pubmed/17659998)
27. Wang H, Suh JW, Das SR, et al. **Multi-atlas segmentation with joint label fusion.** *IEEE Trans Pattern Anal Mach Intell* 2013;35:611–23 [CrossRef](http://dx.doi.org/10.1109/TPAMI.2012.143) [Medline](http://www.ncbi.nlm.nih.gov/pubmed/22732662)
28. Wang H, Das SR, Suh JW, et al; Alzheimer’s Disease NeuroimagingInitiative. **A learning-based wrapper method to correct systematic errors in automatic image segmentation: consistently improved performance in hippocampus, cortex and brain segmentation.** *Neuroimage* 2011;55:968–85 [CrossRef](http://dx.doi.org/10.1016/j.neuroimage.2011.01.006) [Medline](http://www.ncbi.nlm.nih.gov/pubmed/21237273)
29. Dice LR. **Measures of the amount of ecologic association between species.** *Ecology* 1945;26:297–302 [CrossRef](http://dx.doi.org/10.2307/1932409)
30. Crum WR, Camara O, Hill DL. **Generalized overlap measures for evaluation and validation in medical image analysis.** *IEEE Trans Med Imaging* 2006;25:1451–61 [CrossRef](http://dx.doi.org/10.1109/TMI.2006.880587) [Medline](http://www.ncbi.nlm.nih.gov/pubmed/17117774)
31. Kerchner GA, Boxer AL. **Bapineuzumab.** *Expert Opin Biol Ther* 2010; 10:1121–30 [CrossRef](http://dx.doi.org/10.1517/14712598.2010.493872) [Medline](http://www.ncbi.nlm.nih.gov/pubmed/20497044)
32. Wang H, Yushkevich PA. **Spatial bias in multi-atlas based segmentation.** *Conf Comput Vis Pattern Recognit Workshops* 2012;2012: 909–16 [Medline](http://www.ncbi.nlm.nih.gov/pubmed/23476901)
33. Flores GS, de Haan G, Jasinschi R, et al. *Automatic Segmentation of Hippocampal Substructures* [master’s thesis]. Eindhoven: Technische Universiteit Eindhoven; 2012
34. deFloresR,LaJoieR,LandeauB,etal.**EffectsofageandAlzheimer’s disease on hippocampal subfields: comparison between manual and FreeSurfer volumetry.** *Hum Brain Mapp* 2015;36:463–74 [CrossRef](http://dx.doi.org/10.1002/hbm.22640) [Medline](http://www.ncbi.nlm.nih.gov/pubmed/25231681)
35. Wisse LE, Biessels GJ, Geerlings MI. **A critical appraisal of the hippocampalsubfieldsegmentationpackageinFreeSurfer.***FrontAging Neurosci* 2014;6:261 [CrossRef](http://dx.doi.org/10.3389/fnagi.2014.00261) [Medline](http://www.ncbi.nlm.nih.gov/pubmed/25309437)
36. Desikan RS, Se´gonne F, Fischl B, et al. **An automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest.** *Neuroimage* 2006;31:968–80 [CrossRef](http://dx.doi.org/10.1016/j.neuroimage.2006.01.021) [Medline](http://www.ncbi.nlm.nih.gov/pubmed/16530430)
37. Klein A, Tourville J. **101 labeled brain images and a consistent human cortical labeling protocol.** *Front Neurosci* 2012;6:171 [CrossRef](http://dx.doi.org/10.3389/fnins.2012.00171) [Medline](http://www.ncbi.nlm.nih.gov/pubmed/23227001)
38. Pluta J, Yushkevich P, Das S, et al. **In vivo analysis of hippocampal subfield atrophy in mild cognitive impairment via semi-automatic segmentation of T2-weighted MRI.** *J Alzheimers Dis* 2012;31:85–99 [CrossRef](http://dx.doi.org/10.3233/JAD-2012-111931) [Medline](http://www.ncbi.nlm.nih.gov/pubmed/22504319)
39. Mueller S, Yushkevich P, Wang L, et al. **Collaboration for a systematic comparison of different techniques to measure subfield volumes: announcement and first results.** *Alzheimer’s & Dementia* 2013;9:P51