2005-005
Technical Memo

MSc Machine Vision Course : Practicals

N.A.Thacker and P.A. Bromiley

Last updated
10 / 2 / 2005

WWW.TINA-VISION.NET

Imaging Science and Biomedical Engineering Division,
Medical School, University of Manchester,
Stopford Building, Oxford Road,
Manchester, M13 9PT.

MSc Machine Vision Course : Practicals

Neil Thacker, Paul Bromiley

Contents

Introduction

The following pages form the basis of a 10 part practical in machine vision techniques using the **TINA** machine vision environment.

Chapter 1

Introduction

1.1 Introduction and Aim

The aim of this practical session is to gain familiarity of the TINA environment (it does not form part of your assessment). It is expected that the practical should take no longer than 90 minutes.

A useful source of information about TINA is the website,

`http://tina-vision.net/Tina`.

Of particular use is the 'User Guide',

`http://tina-vision.net/docs/technical/user_guide/`

which covers how to use the tools in TINA. Another invaluable source of information is the online code browser,

`http://tina-vision.net/software`

From here you can view any of the code in the TINA libraries. This is particularly useful for learning how a tinaTool button works.

1.2 Getting started

The following is a description of how to get started with TINA in the Niac lab.

- Log onto a machine using the account name and password provided.
- Enter the practical directory. You need to change into the correct directory for that weeks practical. The directories are named after the practicals and are also numbered. So this weeks directory is called '1.Intro'. You can enter this directory with the command `cd 1.Intro`. If you are unfamiliar with UNIX commands speak to the teaching assistant and they can provide you with a list of useful commands.
- Starting TINA. To start TINA you need to type `./tinaTool -r &`. This should cause a TINA tool window (or more probably) to appear on your screen.

1.3 Practical

You will be given a demostration of some basic functionality of the software to give you an idea of what to expect. You then need to familiaries yourself with the basics ready for the laboratory sessions during the rest of the course.

Have a look at the introductory sections in the TINA user guide on the web regarding file I/O and Tv displays. In addition you will need to refer to the sections describing the image calculator " *Imcalc Tool*".

Once you have started TINA you will be confronted with a series of windows. One of these is the *mono tool* which enables you to load images. The other is the *imcalc tool*. There are also two Tv displays, one of these is used to display the image the other is used for display graphs.

Load the `house.l` image into the monotool and `push` the image onto the image stack and displaying it in the Imcalc Tv (hint: if it doesn't appear automatically use the Tv `int` button or you may have to install it with the `install` button).

You are now ready to start using the system for image analysis. Explore the functions in the image calculator using TINA in conjunction with the online user guide. The software also contains built in "help" buttons on many parts of the interface which provide a short form of user documentation including the parameters each function requires. Make sure you master the the following tasks;

- Display profiles and histograms from selected regions of the image. Use these to try and estimate the the level of image noise from the plots you see.

- Load multiple images onto the stack, around which the imcalc tool is built (i.e. the right house image `house.r`). Try adding, subtracting and multiplying images. Try shifting and adding them together.

- Scatterplot the two house images using the `scat` button. Use the `sqrt` to make the image visible. Select a region and inverse scatter, (`iscat`, to recover the regions which share these grey-levels. The more adventurous students may wish to try to use the online code browser to see what is going on.

Chapter 2

Fourier Transforms and Image Noise

2.1 Introduction and Aim

The aim of this practical session is to increase your familiarity with the TINA environment (it does not form part of your assessment). This will be done using the example of Fourier image processing, in particular the convolution theorem, which you should have covered in the first week of web directed reading material. It is expected that the practical should take no longer than 90 minutes.

2.2 Getting started

Login to the system as described in Chapter 1. The TINA executable `tinaTool` for this session can be found in the directory `2.Fourier`.

From the main tool you will need to:

- Start up a `mono` and `imcalc` sub-tool. Load an image into the mono tool of TINA, (e.g.: one in ./house).
- Start a new Tv window and install it as the Imcalc Tv. (You can install other Tv's as required.)
- Push the image onto the image stack and display it in the Imcalc Tv (if it doesn't appear automatically) use the Tv `init` button).
- Now install a graph Tv.

2.3 Practical

Refer to the online manuals for use of buttons (such as `fft` and `rmdc`) in the `Imcalc Tool`. Construct the FFT of an image selected from `images` directory and see if you can relate it's main features back to the original image. Notice that the result is in the form of a complex image, as it must be for a Fourier transform. This is shown by the `Image type` cast bar, which is set to the image type on completion of a calculation, and can also be used to re-cast the data (so that a complex image can be converted to and from two floating point images).

You will encounter the first problem associated with image processing, although the software has automatic data windowing, the dynamic range of computed quantities prevents simple visulalisation of unmodified result. You may have to remove the DC component using the `rmdc` button or select a region of interest in order to see any structure, there are also many non-linear processes which can be performed which will also help.

Try barrel shifting the FFT and the applying the inverse FFT, what do you see and why?

Use the `Create Tool` to generate Gaussian random noise on each pixel and the add this to the original image (the `noise` button generates and image of grey levels with a zero mean Gaussian distribution and the `noise` variable controls the variance).

Design a systematic set of tests to investigate the behaviour of the real and imaginary components of the FFT of the image for various additive noise levels. Use the image histogram and the `noise` button to estimate the variance of the image for different levels of noise.

Attempt to find answers to the following questions:

- Do you see any correlation between the level of added noise and the noise in the Fourier domain? Try investigating this by plotting a graph of FFT noise against original image noise for various values of additive noise.

- Do your results depend on the data in the original image? (You can use the `Create Tool` to generate a few alternatives such as the checkerboard).

- Given that you know how much noise there is in the noisy FFT image can you find a way of estimating this without subtracting the original image?

- If you were to apply a filter to a Fourier domain image (as described by the convolution theorem) what would be the consequences in terms of the noise on the inverse? (Start by thinking about one sinusoidal component).

Chapter 3

Erosion and Dilation

3.1 Introduction and Aim

In this practical you will be gaining familiarity with the processes of binary thresholding, skeletonisation and morphology.

3.2 Getting started

Login to the system as described in Chapter 1. The TINA executable `tinaTool` for this session can be found in the directory `3.Eros`.

Setup TINA with `mono` and `imcalc` tools together with Tv's and graph display.

3.3 Practical

The `images` directory contains an image from an automatic karyotyping system, *karyo.aiff*. This contains several chromosomes on a white background. In an automatic system the first problem is to locate these chromosomes accurately prior to classification. Today we will attempt to do this via the process of **binary thresholding** and **skeletonisation**.

Load the *karyo.aiff* image into the image calculator. By examination of the range of grey levels within the background and chromosome data determine a suitable threshold for binarisation of the image. This can be achieved by casting the image to a binary with an appropriate choice of `threshold` (all values above the threshold will be set to 1 and the remainder to 0).

Now apply the binary skeletonisation `skel` and examine the results carefully. The `skel` button applies the standard algorithm as described in Gonzales and Wintz. You will probably find that although the image can be well segmented and binarised in places, in others the extracted chromosomes become fragmented. Use the `erode` and `dilate` functions to form open and closing operations (as described in the lectures). The implementation of these operations in TINA is as circular grey level morphological operators and you can apply these to either the original or binary image. The output is always a grey level image but an equivalent binary operation can be obtained by appropriate binary thresholding of the output. Can you make the final skeletonised result more reliable (i.e. eliminate gaps) using these processes?

The second set of data for this practical is the image of a retina. Here we are interested in extracting the blood vessels in the image as completely as possible. Repeat the procedures you have developed for the karyotype images with the `retina.aiff` image.

Finally, apply the grey level morphological operators to the familiar house image at a range of scales. You should find that by performing appropriate sequences (as described in your lectures) it is possible to eliminate specific scales of feature.

- What is the common picture medium that displays similar visual characteristics to images which have been processed with a morphological filter?
- Is it possible to perform reliable morphological filtering on either or both of the medical images?
- What are the main charateristics of images that you think are suitable for reliable application of morphological filters?
- Are there any properties of images or image contents which you think should be avoided as input data for such filters?

Chapter 4

Convolution & Deconvolution

4.1 Introduction and Aim

In this practical you will be gaining familiarity with image convolution.

4.2 Getting started

Login to the system as described in Chapter 1. The TINA executable `tinaTool` for this session can be found in the directory `4.Decon`.

Setup TINA with `mono` and `imcalc` tools together with Tv's and graph display as in previous practical.

4.3 Practical

Load the *house.l.aiff* image from the `images` directory.

Apply the `gauss` function in the image calculator to the image. Remember you have to choose a suitable width using the appropriate parameters in `imcalc params` (Hint: use the `help` button to identify the parameters that `gauss` uses). Use a `prof` graph to view the affect on image structure (such as edges) and the `hist` and `noise` functions to see how it affects image noise. Repeat this for various scales of smoothing starting at a kernal width (**sigma**) of 1 and increasing to 5 (you must also remember to adjust the **range** parameter to accomodate this larger filter). Don't make the kernel too big as the execution time can increase dramatically.

Use the `gauss` function to enhance the edges of the image. Do this using the difference of Gaussian (DoG) approximation to the Laplacian. Can you produce a binary image showing the edge locations you have found?

Repeat the above in the frequency domain, remembering that convolution in one domain is multiplication in the other. You will need to construct an image of the Gaussian kernel of the same size as the original image (Hint: use the `delta` in the imcalc `create tool`).

Can you see any appreciable difference between the results? [1].

Finally attempt a Fourier deconvolution of the smoothed image (read carefully the section in the manual regarding division using ”imcalc”).

Attempt to answer the following questions;

- How do results compare: firstly for Fourier convolved images (at a range of scales) and secondly for spatially convolved images?
- Can you explain why this might happen?
- What are the consequences for the application of deconvolution to tasks such as image deblurring?

---

[1] If you subtract the images remember there may be a normalisation to consider

Chapter 5

Noise Estimation and Removal

5.1 Introduction and Aim

The aim of this practical session is to gain familiarity with techniques for noise estimation and removal.

Add various amounts of noise to the image as you have done in previous practical (i.e. use the random image generator in the `create tool`). Use `noise` button in the `imcalc tool` to estimate the level of image noise. Test the technique on several images until you can recover the value you have added reliably and feel you can trust the technique. You may feel you need to modify the basic technique in order to get accurate results, feel free to do so.

Experiment 1

Apply the tangential filter `tsmooth` to the image and estimate the reduction in image noise. This filter smooths along the tangential direction to local edge structure in each pixel. Again, repeat this for various noise levels.

- You should be able to determine an approximate constant factor of reduction. What is it?
- Compare the result with Gaussian smoothing; where are the largest differences between the resulting images and why?

Experiment 2

Use the `create Tool` to generate an image with uniform random noise on each pixel. Threshold the resulting image to obtain a binary image with 10 percent zero values. Multiply an image by this mask image. This will generate a simulated effect of pixel drop out.

- Now apply the median filter and again assess the quantity of remaining pixel noise. Repeat the experiment for various percentages of drop out.
- Plot a graph of the number of remaining drop out pixels as a function of the percentage of simulated drop out. At what point does the algorithm begin to fail to remove all drop out noise?

Experiment 3

Finally we will look at rank order filtering (`rank`) which has been suggested as a pre-processing step for stereo algorithms. Load in both stereo images of the lego house (one at a time) into the image calculator.

- Try to estimate the grey level difference between several pairs of selected corresponding locations in the scenes.
- Make careful notes of the locations you select and the results, you will need them later.
- Express your results as a z-score (ie: the difference divided by the estimated image noise).

Apply the rank filter to both images, the `range` parameter should be left at 5 (which specifies an $11x11$ processing window). You will need to set the `precision` parameter to approximately 4 times the estimated level of image noise. Try to pick a sensible level that eliminates noise but retains discriminable features.

- Measure the fractional difference in grey level for the same correspondences as measured previously?
- Express your results as a z score (ie: difference divided by estimated rank filtered image noise).

Can you say anything regarding the apparent similarity of grey levels for equivalent locations in the original images and the rank filtered images?

Is this a sufficient measure of performance for selecting a suitable preprocessor for a stereo algorithm?

Chapter 6

Deformable Templates

6.1 Introduction and Aim

The aim of this practical is for you to gain familiarity with deformable model based techniques.

6.2 Getting started

Login to the system as described in Chapter 1. The TINA executable `tinaTool` for this session can be found in the directory `6.SROI`. Execute it with **tinaTool -f aorta**.

6.3 Practical

In this practical we shall be assessing the performance of the deformable template when presented with the task of tracking the outline of the descending aorta in an MR sequence of 20 images of the heart. As there is no ground truth data available for the dataset we shall concentrate on the reproducibility of the technique under changes in parameters and also against human operators.

First the model must be trained. You will have to hand outline some images in order to train the system. To do this select `r axis` from the `markup` menu and specify the axis grid. Subsequent points are selected according to the blue radial line. You will not need not to include again the bounding points which have already been specified from the axis. Change the `File name base` to a unique name and output the file. When done click `Make and Output`. Select the next image you wish to mark up (every other image will be sufficient). Reset the mark up by selecting `r axis` from the `markup` menu again and continue.

The next step is to build the model itself. To do this you need a file that lists all of the model files and their relative directory, this should have been automatically constructed and will have the extension .blt. For example if you had named you files *aortamodel0, aortamodel1...aortamodel10* in the directory *./tmp* then the file you create would look like this;

/tmp/aortamodel0
/tmp/aortamodel1
.
.
/tmp/aortamodel10

When this is done ensure that the `File name base` is set to the name of your list file and hit the `Build and Output PCA` button. You now have a PCA model of this dataset.

In order to use the model you should load the mean model `.mml` file and hit `show`. Load the PCA and hit search in order to locate the aorta outline. Hit `show` to redraw the result (note, in some instances the mean model is sufficiently different from the data that the system is unable to find the global minima. In such cases feel free to either move the mean model orchange some of the other model parameters.)

6.4 Evaluation

6.4.1 Experiment 1

Using the images which were not used for construction of the model attempt to search for the aorta outline. Perturb the start location (`tx` and `ty`) of the mean model by a few pixels and repeat the process. Compute the pixel differences between these results. Repeat this analysis for all the missing model images. What do you estimate as being the reproducibility of the generated `roi`'s?

6.4.2 Experiment 2

Using the images from which the missing models were taken attempt to search for the outline. Reduce the number of model modes (`mdl modes` and profile modes `prf modes`) and see what effect this has on the reproducibility of the technique. See how far the model center can be moved before the localisation fails.

6.4.3 Experiment 3

With any `smartROI` outline showing hit the `Outer to ROI` button. Now use the `roi` button in the `imcalc tool` to segment the image. Doing this for other outlines provides images which may be compared using subtraction and image histogramming to count pixel differences. In this way you are able to score the performance of the algorithm when used to extract the boundary of a given image. Use the `poly roi` to hand draw around the aorta of the missing data images. Ask another person who has not seen your outline to do the same. Compare the reproducibility of the resulting `roi`'s. How do they compare to your PCA results?

Some notes

- The search ranges reset to zero following output of a sampled profile, preventing search unless the parameters are reset.
- The search does not work unless the search parameter dialog is open.
- The mean model can only be loaded after the PCA model.
- The profile model is set to 0 parameters by default and no profile modes are displayed in the parameters dialog.
- The markup process can only be restarted from the mouse option menu. The restart (middle button) only resets to the second landmark but does not allow respecification of the baseline.

Chapter 7

MR Image Segmentation

This session forms part of your assessment

7.1 Introduction

The exercises are to be carried out using the TINA image environment. A moderate degree of collaboration in carrying out the work is acceptable but reports should be entirely your own. You can expect to have to draw upon both course work and previous practical experience. In each case you should explain the objective of the exercise, outline the combination of processes you used, explain the results you obtained the scope (ie general usefulness) of the method and try to address any specific questions raised. You do not need to explain the basics of TINA. Tackle the following problem using TINA and produce a write-up of approximately 3-4 pages.

7.2 Practical

The software you should use for this exercise is in `7.NMRSEG`.

The image calculator contains two buttons, `scat` and `iscat`. The first of these produces a 2D `float` scattergram image from the selected region of interest (ROI) of a `complex` image, where the x axis is the real pixel value (0-256) and the y axis is the imaginary pixel value (0-256).

The `iscat` button takes a `float` scattergram and a `complex` image and generates a new `float` image by replacing the image pixels with the corresponding value within an identified region of interest (ROI) of the scattergram. Using these two buttons it is possible to perform a dual valued (`complex`) image segmentation.

Read in the `dual_echo1` and `dual_echo2` images from the `images` directory and form a composite complex image in the image calculator (remember to normalise the image to occupy the available grey level range used in the scattergram). This can be as simple as using the two images as the real or imaginary components directly, but you may get better results from constructing alternatives such as sum or difference images. Produce a scattergram of the complex image for regions containing

- mainly CSF (dark pixels within the cranial cavity).
- mainly white matter (central regions of brain tissue).
- mainly grey matter (the peripheral regions and folds of the brain)

Try to get a feel for where the various tissue types appear in the scattergram.

Now, using the `iscat` process as a starting point, attempt to produce three segmentations of the image for the various identified tissue types.

Describe the following in your report:

- A specification of the objectives.
- The motivation for the approach.
- The final version of your segmentation technique.
- A brief description of the results including images.
- The shortcomings of the process.
- Any suggestions you might have for improving the basic approach.

Chapter 8

Texture Recognition

8.1 Introduction and Aim

The aim of this practical is for you to gain familiarity with techniques in texture recognition and the problems associated with it. In particular you will be using Gabor filters in an attempt to identify textured regions in an image.

8.2 Getting started

Login to the system as described in practical 1. The TINA executable `tinatool` for this session can be found in the directory `8.Text`.

8.3 Practical

Gabor filters are often presented a solution to texture recognition problems in machine vision largely because of support from psychophysical experiments. The 2D Gabor function is a harmonic oscillator, composed from a sinusoidal plane wave at a particular frequency and orientation, within a Gaussian envelope.

Using the `fgabor` function in the `create tool` generate Gabor filters. This function generates filters in the Fourier domain so you will need to run `ffti` to see the spatial equivalent. Adjust the parameters `gb_k`, `gb_b` and `theta` and see how the resulting filter changes.

Load the *broadatz.aiff* image in the *image* directory onto the stack. As convolution is essentially a pattern matching process, it is possible to build filters which respond well to particular types of texture structure of fixed scale and orientation. Attempt to extract the different regions in this image using your knowledge of the way in the which the parameters control the kernels to match them to the scale an orientation of the different textures. Create Gabor 'tuned' to a type of texture shown in the image. Apply the filter to the image as a multiplication in the Fourier domain, `*`. Transform the result back into the spatial domain using `ffti` and calculate the magnitude of the response. This is done by taking the `sqr` of each component of the complex result (cast to `float`), summing `+` and computing the `sqrt`. You should see some areas where the filter gives better responses than others. Try to assess the sensitivity of the response to parameter change by seeing how much the parameters can be varied before the high response is lost.

Brodatz textures are particularly suited for texture recognition, there are no illumination artifacts and no perspective effects distorting the appearance of the texture. These issues and the problem of unpredicatble responses from the texture filters at boundaries, are the main obstacles to using such filters in general applications. If you have time try using the *house.l.aiff* image. Can you find a filter tuned to extract the window shutters? What problems might arise using this technique if the object was viewed from a different direction?

Chapter 9

Stereo Vision

This session forms part of your assessment

The exercises are to be carried out using the TINA image environment. A moderate degree of collaboration in carrying out the work is acceptable but reports should be entirely your own. You can expect to have to draw upon both course work and previous practical experience. In each case you should explain the objective of the exercise, outline the combination of processes you used, explain the results you obtained the scope (ie general usefulness) of the method and try to address any specific questions raised. You do not need to explain the basics of TINA and should avoid giving descriptions based only upon use of the interface. Tackle the following problem using TINA and produce a write-up of approximately 3-4 pages.

9.1 Practical

The software you should use for this exercise is in `9.STEREO`.

The goal of this practical is to use the TINA vision system to perform edge based stereo and wireframe reconstruction on the sets of images provided in the *images* directory, (*shaft_assembly, saucer, big_crystals*). The algorithm used is a variant on the PMF algorithm (as covered in the lectures) and produces data which is typical of edge based approaches. More details are given in the Tina Users Guide, which can be found on the tina web site.

Start the tool using `tinaTool -r` to get the stereo setup. Load and view the first pair of images. Ensure that the `.cam` file have been loaded too. Compute the edge maps for these images using the `canny` button. You need to rectify the edge data using the `rectify` button before you compute the depth with `stereo`. Lines and curves can be fitted to the disparity data using `init geom3` and geom 3. You can control all of these algorithms using the parameters in the 3 `Params` dialogs at the bottom of the `edge tool` (the help system will give you assistance in using these).

This vision system is intended to provide data suiatble for location of objects in a robot control task. Perform experiments which enable you to discuss the following in your write up:

- The stability of the algorithm to parameter adjustments. For example, what effect do changes in edge threshold have on the quality of the results? How do changes in disparity limits change the outcome of the matching?
- The major strengths and weaknesses of the reconstructed stereo data and the consequences of this for automatic model matching.
- How you might attempt an automatic evaluation of such algorithms? (i.e. what data you would need and what things you would need to measure)

Describe the following in your report:

- A specification of the objectives (What?).
- The motivation for the approach (Why?).
- The final version of your evaluation of the stereo algorithm.
- A brief description of the results including images.
- The shortcomings of the methods.
- Any suggestions you might have for improving the basic approach.

Chapter 10

Object Recognition

10.1 Introduction and Aim

This session will cover a slightly more advanced machine vision topic than grey level image analysis, object recognition. The TINA system will be used to match polygonal boundaries (extracted from images by the Canny edge detector) using the technique of Pairwise Geometric Histograms, as described in the lectures. Here a theory is suggested that the use of a complete description of data with the desired levels of invariance, matched with an appropriate statistical method, forms an optimal solution to the recognition of arbitrary objects. In this practical we will attempt to evaluate the performance of the recognition system subject to changes in the shape representation.

10.2 Getting started

Login to the system as described in practical 1. The TINA executable `tinatool` for this session can be found in the directory `10.Pair`.

10.3 Practical

Good descriptions of this tool can be found in the users guide in the section describing the `pairs` tool.

Execute this program using `tinaTool -r` to invoke the stored replay. This will initialise the various parts of the tool for graphic display and load in an example set of models and scene data. The following sequence of button presses will attempt to identify the match outline in the cluttered scene.

- `match lines` : match the stored line histogram's to the scene
- `segment model` : display the sections of data consistent with the chosen `Model Name`
- `locate models` : locate all examples of the chosen `Model Name`

- `graph` : display the radial error from the probabilistic Hough Transform

This will automatically load the polygonal models *as?.poly* stored in the local `dinosaur` directory and will build a data base of corresponding geometric histograms with the parameters specified in the `pairs` tool.

You can examine the typical appearance of a histogram by selecting the `pick (geom)` option in the pairs tool and picking a line from the scene displayed in the `mono` tool using the mouse. Executing `histograms` will generate the corresponding PGH on the top of the stack in the image calculator (you may need to `init`' the `Tv` periodically to refresh the imcalc `Tv` display with the new data).

You can match individual features of the scene to the object fragments stored in the PGH data base using the `match line` routine. Select a few lines to see how well the default options appear to work.

You can run an entire scene analysis using the `match scene` routine. The `segment` button will now allow you to identified the fragments of the scene associated with the **Model Name** parameter. Quantify the errors for each model in the data base *as1-as5* in a table. The table should contain the number of line fragments in each object which have been correctly and incorrectly labelled.

Now clear the data base and reload the models with a different selection for the histogram type (directed). This form generates a larger histogram representation than previously such that the set of histohrams for each object comprise a complete representation of the shape.

Try to relate what you see in the histogram now to what you saw with the `rotate` parameter. Re-run the scene matching and evaluate the recognition performance once again.

What is the fractional improvement in labelling reliability? Is this improvement statistically significant?

Chapter 11

Time Course MRI Data.

11.1 Software

The software you are using supports several methods for the analysis of time course MRI data. We will start by gaining familiarity with some of them.

11.2 Aims

The objective of this practical is to develop an understanding of the use of Monte-Carlo methods in the quantitative assessment of analysis techniques.

11.3 Experiment 1

Run the macro files associated with each of the following so that you know how to process the associated data.

11.3.1 FMRI

FMRI is the acronym given to the identification of brain activity in MR image data via a complicated mechanism of stimulated blood flow changes.

Configure the TINA system for by executing 'tinatool -f fmri' from the UNIX command line or running tinatool1.sh. The 'Stimulus' correlation function must be 'plot'ed in the 'imcalc graph' by pressing the appropriate button. Equivalent statistical measures to many used in the literature can be selected from the 'Sequence' menu and used to 'compare' the temporal data behaviour. An image of the degree of correlated response to the experimental stimulus will be displayed as an image in the imcalc Tv, which can be interpreted as a null hypothesis test of the original data being consistent with noise.

11.3.2 T2* Brain Perfusion

Perfusion is the name given to the delivery of blood and its constituents to tissue. A contrast injection can be given which makes this process visible in MR images. A gamma function can be fitted to the concentration time curves in order to extract parameters associated with perfusion.

Configure TINA with the command ‘tinatool -f perf’ or running tinatool2.sh. Generate a binary region of interest in the imcalc tv. and compute selected fit parameters ”gamma fit”. Primary parameters are relative blood volume (CBV) and bolus arrival time (TTM). The results can be displayed as a parametric image in the imcalc tv. Having run the initial analysis, images for each of the fit parameters can be generated via the choice dialog switches.

11.4 Experiment 2, Assessed Practical

Select one of the above methods for your experiments. You should refer to the appropriate section of the Tina documentation for this technique before continuing.

Using methods you have already learned on this course, estimate the amount of image noise in the input images. Remind yourself of how to generate random noise images with a similar quantity of white noise.

Take the original image data, and by adding suitable amounts of random noise (to all input images), determine the change in output data caused by additional random noise similar to the measured image noise.

Using the scatterplot methods construct histograms of measured value against the change in that value seen due to noise (Bland-Altmann plots). You will need to remember to scale data appropriately, this should be done so that the scaling on each axis is known.

Summarise and explain your findings, illustrating with suitable images and figures in a report of up to 5 pages.

Try to answer the following questions;

What are typical expected levels of noise in the output data as a percentage of the computed quantity?

Does random uniform noise produce uniform noise on output quantities?

How might the behaviour of data limit the use of these outputs?