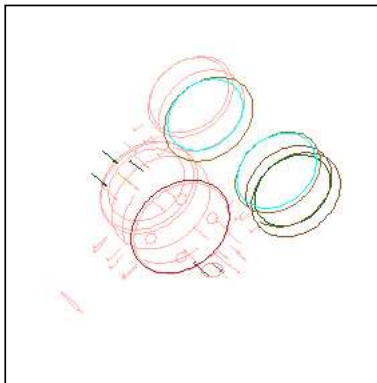


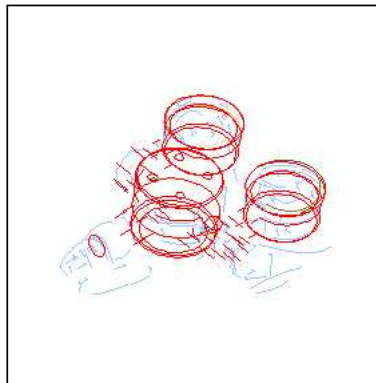
TINA Demo 002: The Stereo Vision Tool

neil.thacker(at)manchester.ac.uk

Last updated
30 / 10 / 2008



(a)



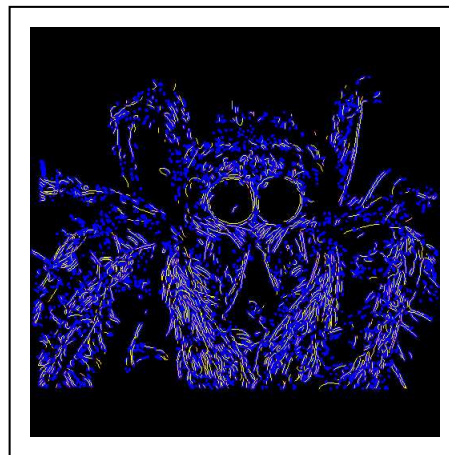
(b)



(c)



(d)



(e)



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

TINA Demo 002: The Stereo Vision Tool

N.A. Thacker.
Dept. of Medical Biophysics
Imaging Science and Biomedical Engineering Division
Medical School, University of Manchester
Manchester, M13 9PT, UK
`neil.thacker(at)manchester.ac.uk`

1 Introduction

This directory contains software for wire-frame object recognition (localisation) originally developed at AIVRU in the University of Sheffield. The stereo vision demonstration summarises the most recent methods for stereo image analysis developed by the TINA group over the last 15 years. It includes methods for calibration, stereo matching, 3D reconstruction, model matching and object localisation. The main intention is to provide working examples which illustrate results in published papers. The software will run with a minimum of user interaction, in order to give a feel for what is possible and data quality. The software can be used to extract edge based stereo data from a pair of calibrated images (see calibration demonstration). This data is then used to generate 2D and finally 3D data which is matched to a wireframe model.

Typical problems that the approach is designed to cope with include inaccurate and imperfect data segmentation, scene clutter and object occlusion. Current limitations of these techniques include; an inability to coherently discriminate between different objects and the need for manual construction of the initial 3D model. These problems are being addressed in other work (see documents relating to pairwise geometric histograms). However, techniques shown here however have been demonstrated to be more than adequate for robotic control tasks, such as pick and place, in restricted environments. We continue to develop them when the opportunity arises. Used in conjunction with the Tina Users Guide (2005-002) we believe there is sufficient information so that you can substitute your own data sets for processing.

Auto-calibration provides a robust solution to the problem of maintaining an accurate estimate of absolute camera geometry. So-called uncalibrated stereo is not-necessary if prior information is integrated over time using appropriate estimates and updates of covariance matrices, as here. The degree of flexibility in the camera model is selectable and include radial distortion terms. The error scores are robust estimators which have been shown to be able to deal with more than 30% outliers with initial rotation estimates within 30 degrees of the true value (this is well within the requirements of any practical vision system as the verge angle is normally within 5 degrees of a symmetric verge at 15 degrees and the number of outliers from the corner matcher is generally much better than a few percent). Scaling intrinsic parameters can be determined from metric 3D data before attempting auto-calibration, thereby supporting full euclidean reconstruction.

2 Installation

In order to use the TINA stereo vision tool, you must already have compiled and installed the TINA libraries. See the TINA website

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

for instructions.

The TINA stereo vision tool project can be obtained either from the demos page of the TINA website or from the TINA CVS repository using the commands

```
cvs -d :pserver:anonymous@cvs.tina-vision.net:/home/tina/cvs login
(press enter when prompted for a password)
cvs -d :pserver:anonymous@cvs.tina-vision.net:/home/tina/cvs co tina-projects/vision_tool
```

This should check out the following files to the current directory

README: asci versions of this document.
Makefile: Executed project Makefile.
Makefile.linux alternative linux Makefile.
Makefile.solaris: alternative solaris Makefile.
002_vision_tool.pdf: this document.
tinaTool.c: main tool specification software.
images: subdirectory of example data.
tlbaseHist_tool.h : function prototypes for library tools
tlvisCalib_tool.h : as above.
tlvisSmm_tool.h : as above.
???.cls : tina replay files (human readable asci)

In order to get started you must first build the tinaTool. Copy either the Linux or Solaris to alternative to Makefile (the default is already Solaris). This Makefile may require some manual editing: it assumes that you installed the TINA libraries in the /usr/local/Tina5 directory, and that you are using the Xview widget set. Open the Makefile in an editor, set the TINA-X-GUI variable to specify the widget set you wish to use, and edit any paths that do not apply for your system. Finally, type “make” to build the tinaTool executable.

To run the vision tool you can type one of the following,

```
./tinaTool -f imcalc  
./tinaTool -f pmfgeom  
./tinaTool -f corrgeom  
./tinaTool -f corner  
./tinaTool -f grid  
./tinaTool -f match
```

which executes a series of button presses stored in associated “.cls” files. See below for details.

3 Usage

3.1 Graphics

Refer to the Tina Users Guide for information regarding installing and operating general purpose graphics panels (Tv’s). The TINA vision demonstration makes use of a variety of Tv’s for image display. These can be interchanges and replaced as necessary, for example; Click on “New Tvtool” in the top level tinaTool window to start a new Tvtool to display this graphics stream, then click on “left” in the Tv list of the Stereo tool, and finally “install” in the Tvtool window. The Tvtool will now display any “left” stereo images loaded into the Stereo tool. The macro’s will generally perform these steps for you.

3.2 Image Input

In order to use TINA with your own data , some knowledge of the image format is required. Generally you should convert any images to a grey scale pgm (portable grey level map) format in order to load. Derived image results should be stored as the Tina ”aiff” if you wish to maintain precision and store co-ordinate space details. In order to display the header information an image can be pushed from the relevant tool into the Icalc Tool and information displayed using the “info” button. See User Guide for Details.

The stereo tool supports input of 2 images as a “left” “right” pair, on the basis of the provided **Directory Name** and file **Base Name**. Click “input” to load.

3.3 The Image Calculator

A large part of software development is involved with analysing data distributions. One example is in the analysis of errors on computed quantities resulting from noise on input data. A simple example is provided here illustrating the use of TINA for analysis of statistical distributions.

Many algorithms can be analytically evaluated using error-propagation. In cases where this cannot be done they can generally be evaluated by Monte-Carlo. Monte-Carlo is the process of assessing the effects of small perturbations

in the data (typical of the expected noise) on the output. The Imcalc Tool in TINA has been specifically designed to support such evaluation of image processing algorithms.

It is assumed that you have the standard documentation for the Imcalc Tool available (see our web pages). The basic idea is that the tool works like a conventional calculator, but applies the selected processes to whole images. The Imcalc Tool contains what we believe to be a sufficient level of functionality to allow the execution of most standard image processing algorithms.

To run the demo type "tinaTool -f imcalc". This will configure the tool and load a test image into the image calculator. Duplicate the test image on the calculator stack using 'dup'. For many image processing tasks, it is possible to perform the equivalent process (such as an edge enhancement) by pressing a suitable sequence of buttons in the Imcalc Tool. These can be stored using the Tina macro system for automatic application to multiple datasets.

To run a pre-defined edge enhancement replay simply "run" the macro ("imcalc_edge" already specified) from the main Tinatool.

To see the effects of noise on this process, Gaussian random noise can be generated in the Create Tool using 'noise' and added (+) to the duplicate test image (use 'flp' to get it to the top of the stack). Subtracting the two answers (-) gives the propagated noise distribution. This can be viewed using the histogram facility (see user documentation).

4 Basic Edge Stereo (PMF)

The PMF (Pollard, Mayhew and Frisby) algorithm is now a citation classic, known mainly for its relationship to psychophysical experiments. The algorithm here is included mainly for historical reasons as a far more robust algorithm resulted from the work which re-formulated the same principles for hardware implementation (stretch correlation).

To replay the matching sequence first type "tinaTool -f pmfgeom". This will initialise the various parts of the tool, installing the model and scene display Tv's, and read in a set of example data (shaft_assembly2 images and camera files). Then press the following buttons:

Edge Tool

```
canny (detect edges at a scale of 1 pixel)
edge rectify (convert coordinates to a parallel camera geometry)
geom2 (compute 2D representation of edge strings)
```

PMF Stereo Tool

```
stereo (match epi-polar data for extended string segments)
geom3 (project 2D strings into 3D)
threed Tv (init) (display results)
```

This will display the results of the stereo matching and 3D geometric reconstruction in the threed Tv. The edges of the scene can be removed by specifying regions of interest for edge detection (see User Guide). Corresponding edge string matches can be investigated using the appropriate Pick facilities in the Matcher Tool.

The more recent (and more reliable) stretch correlation algorithm can be run by typing "tinaTool -f corrgeom". As stretch correlation requires rectified images (rather than edges) the processing is a little different to PMF. Resulting correspondence information is used to match Canny edges giving data in an equivalent form to PMF (see Tina memo 2003-009).

Press the following buttons:

Edge Tool

```
im rectify (convert coordinates to a parallel camera geometry)
canny (detect edges at a scale of 1 pixel)
geom2 (compute 2D representation of edge strings)
```

Correlation Stereo Tool

```
stereo (match image patches using correlation)
geom3 (project matched edge data into 3D)
threed Tv (init) (display results)
```

The correlation matches improve following iteration, with the number of matches reported in the tinatool dialog panel. This algorithm and the geometry fitting routines were recently over-hauled (2008) as part of an on-going project, giving substantially better reliability, particularly with conic sections (60% \rightarrow 99%). The replays "corgeom4" (see figures (d) and (e)) and "corrgeom5" show some of the data used during evaluation of the changes. Make sure that the current Tv is set to "threed" to display animations with

```
View Tool
make seq
show
```

3D data can be outputted via the stereo tool (RHG) and loaded as an alternative data set for the following demonstration.

5 Matcher

To replay the matching sequence first type "tinaTool -f match". This will initialise the various parts of the tool, installing the model and scene display Tv's, and read in a set of example data (shaft_assembly model, shaft_assembly2 scene and shaft_assembly cliches) from the shaft_assembly directory. To continue with the matching process press the following sequence of buttons;

```
build tables (build the pairwise geometric tables)
smm matcher (match the geometric tables for the scene to those
             for the model and display consistent cliches)
comp trans  (match remaining scene data to 3D model and display
             the new set of matching features)
trans model (compute the transformation of the model to the scene
             and display the model over the scene data)
```

Alternative sets of results (eg: brake) are also provided. These can be loaded using the File/View tool and matched using the above sequence. These tools can also be used for manual and semi-manual matching. A more complete description of the use of this tool can be found in the user documentation.

Modification of this code for development purposes can be done by following the standard procedures outlined in the tinaTool demonstration directory.

6 Model Based Calibration

This technique is mainly intended for semi-automatic construction of view files, via the statistical verification of feature visibility. Following the matcher demonstration continue with the following buttons;

```
Calib Tool
model (initialise data structures and get estimated object location)
model min (adjust camera parameters to optimise the object projection)
model proj (display results, blue/green points fail/pass hypothesis tests)
```

Replay 'match2' illustrates the process for another object. The chrome plated oxygen valve provides the largest challenge for all aspects of the system.

The "model params" dialog menu controls all aspects of the alignment process, and subsequent feature validation (including print output). See user documentation and Tina memo 2006-007 and 2007-011 for more details.

7 Grid Based Calibration

Getting an initial calibration for the stereo cameras is often the largest obstacle to analysing your own data. Stereo images are often calibrated from a simple grid. The main purpose is to determine intrinsic parameters

(focal lengths, centres etc.), often many of these parameters can be specified as reasonable defaults. Though grid calibration can determine the between camera stereo transformation, this can easily be determined otherwise (see below).

Vertex positions for the grid are stored in a simple ascii file and vertices from intersected lines obtained using an edge detector are used to compute the camera parameters. Such a grid is easily constructed on a laser printer, and the calibration file can be constructed with a text file editor.

Initialise the tool using "tinaTool -f grid".

Calib Tool

```
grid      (obtain vertices from edge data)
Tsai      (fast estimation of camera parameters)
IP min    (accurate calibration of independent left and right cameras)
IPS min   (joint calibration of left/right camera system using previous
          stereo calibration and covariance, if present (eg: init covar))
grid geom (optional; set computed grid vertices for display in Threed Tv)
```

Results of the calibration can be visualised in the graph tv_screen, using the r-dist (radial error), epi-err(epi-polar error) and x-err calib plots. Good results can only be achieved on this test data set by allowing the image centres and x aspect ratio to change (use the toggle switches).

The Tsai algorithm is a closed form approximation which provides a suitable starting point for the more principled statistical estimators (IP and IPS min). Calibration results, including covariances, are written out using the "output cam" button.

Once the intrinsic parameters have been determined they can generally be fixed in subsequent calibrations.

8 Auto Calibration

If a grid is unavailable, stereo images can be calibrated using the following method (see Tina memo 1991-002).

There are many myths surrounding stereo calibration, some perpetuated by high citation papers. One such myth is that auto-calibration must be done with 'Essential Matrices' and can only deliver geometry up to an homology. If the camera model is specified using a pair of conventional rotation matrices, each with orthogonal axes, and the intrinsic parameters (excluding focal length) are known, then auto-calibration can deliver Euclidean geometry up to one unknown scale factor, that being the separation of the cameras. This too can be determined if you know one single distance measurement in the reconstructed scene. Lack of knowledge of pixel size can be also be overcome by observing the angle between perpendicular lines in reconstructed data. Fitting the parameters of a conventional representation with a simple brute force optimisation (simplex) actually out performs the more elegant mathematical approaches (see Tina memo 2000-009). There are not many published papers on the technique as it is mathematically unsophisticated and can be described in one sentence. "Adjust the parameters in a two camera system to minimise the image plane epi-polar re-projection of matched points". You can add that it should use a robust measure if you have space for a second sentence. The main omission in the literature is that a calibration technique must deliver parameter covariances if the results are to be used in a practical system. As many closed form solutions are not based upon statistical estimation (eg:Tsai above), this possibility is precluded for many approaches.

The auto-calibration demonstration uses the results of the stereo corner matching algorithm. Replay the user interface by typing "tinaTool -f corner" . This will configure the tools and load in a corner rich dataset (specifically chosen for its corner ambiguity in order to make the matching problem as difficult as possible). Then follow the following sequence in the Corner Tool.

```
corner    (detect corner features using the Harris/Stephens method)
rectify   (compute estimated location in a parallel camera geometry
          according to the current calibration estimate)
stereo    (match corner features according to the displayed Stereo Parameters)
init geom (delete previous 3D geometry data structures)
geom3     (add the estimated 3D locations to the 3D data structures)
```

then perform an 'init' in the threed Tv. This will display the data which can be zoomed etc. in the usual manner. This can be used as a visual check before proceeding.

To calibrate with this data start the Calib Tool. Then

```
get calib    (get the current stereo calibration used in TINA)
corners     (get the image locations of the matched corners)
init covar  (initialise the camera covariance)
Epi min     (minimise epi-polar errors constrained by the prior covariances)
```

This data can be combined with previous calibration results using the previous calibration and parameter covariances. These are loaded from file by the "input cam" button.

Results of the calibration can be visualised using the epi-err function in the graph tv_screen. The new calibration can be passed back into TINA using 'set calib'. The new calibration is applied by re-running the corner matcher from 'rectify' onwards. For the dataset here the number of matcher features will improve by around 10% due to the better epi-polar search.

Auto-calibration makes it possible to compute stereo geometry from hand held images (eg: holiday photographs). The two images must be steady, be taken from approximately the same distance from the subject and with an angle between the cameras of approx 15 degrees. When processing your own data you must first provide a nominal calibration to get the corner matching started. It is essential with this approach that intrinsic parameters (image centres, pixel sizes, aspect ratios and focal lengths) are initialised manually. You can do this via the params dialog menu in the Calib Tool and then set these parameters for use in Tina (set calib). The epi-polar lines can be visualised on the Left and Right Tv's in order to check that the geometry is approximately correct for corner matching and the search range can be extended to compensate for initial misalignment. Once the corner matching process gives a reasonable set of good matches (green crosses) you can proceed to the Calib Tool for parameter estimation.

The ultimate demonstration of auto-calibration is its use for 3D measurement from scanning electron microscope images (you cannot construct a stereo rig for a conventional SEM). "tinaTool -f corgeom2" shows the 3D data for a small pit generated by laser in a metal surface. Press the 'make seq' and 'show' buttons in the View Tool to rotate the 3D data. See also the stereo replay "tinaTool -f corgeom3", for data reconstructed by this method which was obtained with a digital camera while visiting VIE 2008 China (be patient if you re-run the stereo matching as the images are large).