# GeoPIV:

## Particle Image Velocimetry (PIV)
## software for use in geotechnical testing

**White D. J.[1], & Take W.A.[2]**

**CUED/D-SOILS/TR322**

**(October 2002)**

[1] Research Fellow, St John's College, University of Cambridge

[2] Research Fellow, Churchill College, University of Cambridge

# GeoPIV:

## Particle Image Velocimetry (PIV)
## software for use in geotechnical testing

**Cambridge University Engineering Department**
**Technical Report**
**CUED/D-SOILS/TR322**

**D.J. White & W.A. Take**

**October 2002**

## Summary

GeoPIV is a MatLab module which implements Particle Image Velocimetry (PIV) in a manner suited to geotechnical testing. This brief guide describes the practical details of using GeoPIV to measure displacement fields from digital images. In addition, some common pitfalls are described. The performance of the GeoPIV software is summarised, and the references from which further information can be found are listed. The software was written by the Authors during their PhD research.

## 1 Introduction

The GeoPIV software implements the principles of Particle Image Velocimetry (PIV) in a style suited to the analysis of geotechnical tests. This technical report explains how to use the software and summarises the validation procedures undertaken during the development of the software.

PIV is a velocity-measuring procedure originally developed in the field of experimental fluid mechanics, and is reviewed by Adrian (1991). GeoPIV uses the

principles of PIV to gather displacement data from sequences of digital images captured during geotechnical model and element tests. GeoPIV is a MatLab module, which runs at the MatLab command line. The development and performance of the software are described in detail by White (2002) and Take (2002). Concise details are presented in White *et al.* (2001a, 2001b).

The principles of PIV analysis are summarised in Figure 1. The analysis process used in GeoPIV is indicated by the flowchart shown in Figure 2. PIV operates by tracking the texture (i.e. the spatial variation of brightness) within an image of soil through a series of images. The initial image is divided up into a mesh of PIV test patches. Consider a single of these test patches, located at coordinates $(u_1, v_1)$ in image 1 (Figure 1). To find the displaced location of this patch in a subsequent image, the following operation is carried out. The correlation between the patch extracted from image 1 (time = $t_1$) and a larger patch from the same part of image 2 (time = $t_2$) is evaluated. The location at which the highest correlation is found indicates the displaced position of the patch $(u_2, v_2)$. The location of the correlation peak is established to sub-pixel precision by fitting a bicubic interpolation around the highest integer peak.

This operation is repeated for the entire mesh of patches within the image, then repeated for each image within the series, to produce complete trajectories of each test patch.
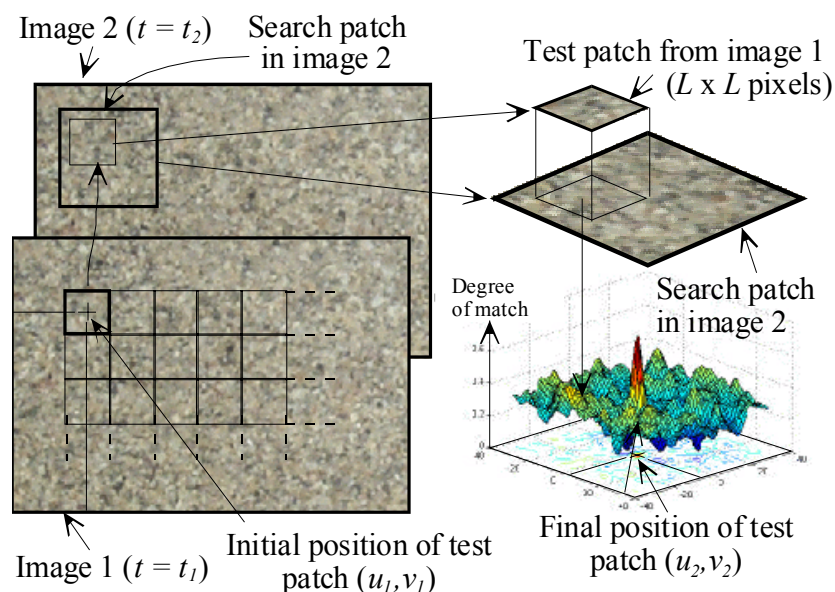


Figure 1. Principles of PIV analysis

Select test patch from mesh in image 1 → Evaluate cross correlation of test and search patch using FFT

Repeat for other test patches in image 1

Use bicubic interpolation to find sub-pixel location of correlation peak

Normalise by cross-correlation of search patch with mask
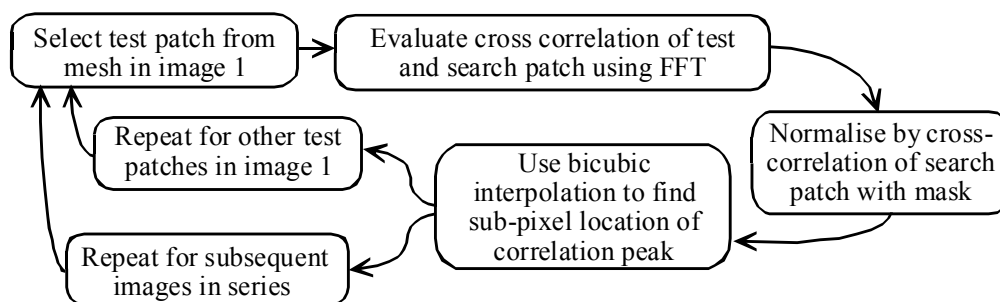
Repeat for subsequent images in series

Figure 2. Flowchart of the GeoPIV analysis procedure

The MatLab module requires two input files (a launcher and an initial mesh file) which are prepared in ASCII format by the user. Simple MatLab scripts can be used to assist the preparation of these input files. The output files are in ASCII format, and can be manipulated by the user in MatLab or a spreadsheet to produce displacement and strain data.

## 2 Software validation

The performance of a measurement system can be assessed by considering the errors associated with accuracy and precision. Accuracy is defined as the systematic difference between a measured quantity and the true value. Precision is defined as the random difference between multiple measurements of the same quantity.

Any deformation measurement system based on image analysis consists of two stages. Firstly, the displacement field between two images is constructed. Secondly, this displacement field is converted from image-space (i.e. coordinates in terms of pixels in the image, or mm on the photograph) to object-space (i.e. coordinates in the observed soil).

The precision of a system depends on the method used to construct the displacement field. Random errors associated with the precision of image-based displacement measurement systems include human error in film measurement, and random errors induced by changes in lighting in centroiding (or 'spot-chasing') techniques.

The accuracy of a system depends on the process used to convert from image-space to object-space coordinates. Systematic errors associated with the accuracy of image-

based displacement measurement systems arise if the spatial variation in image-scale (i.e. the ratio between lengths in object- and image-space) is ignored.

The GeoPIV software is used to construct the displacement field in image-space coordinates. The conversion from image-space to object-space is separate process, and must be carried out subsequent to the PIV analysis. Validation of the GeoPIV software requires the precision of the technique to be established. The accuracy of any resulting measurements depends on the user's technique of converting from image-space to object-space coordinates.

The image-space to object-space conversion process can be carried out by assuming a constant image scale, or by using photogrammetry to establish the image- to object-space transformation more accurately. Taylor *et al.* (1998) and White *et al.* (2001b) present systems based on the principles of close range photogrammetry. White (2002) describes the photogrammetric reconstruction procedure used in the latter system. Take (2002) describes the target location technique used to perform accurate photogrammetric reconstruction, and assesses the accuracy of this system.

The precision of GeoPIV over small displacement increments was initially evaluated by White *et al.* (2001a), and was considered in greater detail by White (2002) and Take (2002). An experimental apparatus consisting of a translating brass container allowed a non-deforming plane of soil to be translated horizontally beneath a rigidly fixed camera. Small known increments of movement were applied to the soil container via a micrometer and the resulting sequence of images was analysed using GeoPIV. The precision of GeoPIV was evaluated by comparing the displacement vectors deduced from a grid of PIV patches overlying the soil. Since the soil translates as a rigid body, the displacement vectors should be identical; the random variation within the measured vectors indicates the system precision. In addition, artificial images were created and tested in a similar manner.

The precision was found to be a strong function of patch size, $L$, and a weak function of image content. An empirical upper bound on the RMS error, $\rho_{pixel}$, is given by Equation 1. Although a larger patch size leads to improved precision, the number of

measurement points that can be contained within a single image is reduced. Larger patches 'smear' the displacement field in area of high strain gradient. A compromise is necessary. The number of measurement points, $n_{points}$, that can be fitted in an image depends on $L$ and the number of pixels within the image. Equation 2 indicates the number of measurements that can be obtained as a function of image width, $W$, and height, $H$, in pixels. Equations 1 and 2 are combined in Figure 3 to show the potential precision and measurement array sizes that can be achieved for various sizes of camera CCD.

The performance of GeoPIV compares favourably with the precision of commercial PIV software used in experimental fluid mechanics, although the processing speed is significantly slower. Christensen *et al.* (2000) report an RMS error of 0.0537 pixels for $L = 64$, when using orthogonal one-dimensional curve fits through the highest integer pixel correlation peak to establish the sub-pixel displacement increment. GeoPIV uses a slower sub-pixel estimator, in which a bicubic interpolation is fitted to the correlation peak. This sub-pixel estimator is considered responsible for the improved precision, but adds a significant computational burden.

$$\rho_{pixel} = \frac{0.6}{L} + \frac{150000}{L^8} \qquad \text{(Equation 1)} \qquad n_{points} = \frac{WH}{L^2} \qquad \text{(Equation 2)}$$
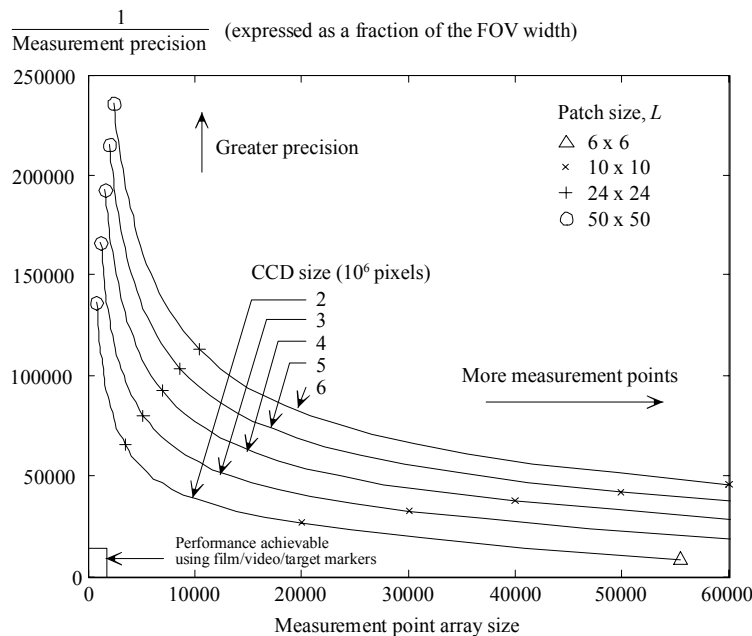


Figure 3. GeoPIV precision and measurement array size vs. camera CCD resolution

# 3 Software usage

Figure 4 shows schematically the steps required to conduct PIV analysis on a series of digital images using the GeoPIV software. The user is required to prepare two ASCII input files.

GeoPIV7_launcher.txt (at the time of writing the latest development of GeoPIV is version 7) lists the images to be analysed and the display parameters to be used during the run. GeoPIV7_mesh.txt contains the coordinates and sizes of the initial grid of PIV patches. This grid of patches is established in the first image of the series and each patch is tracked through the subsequent images. The two input files are formatted as follows.
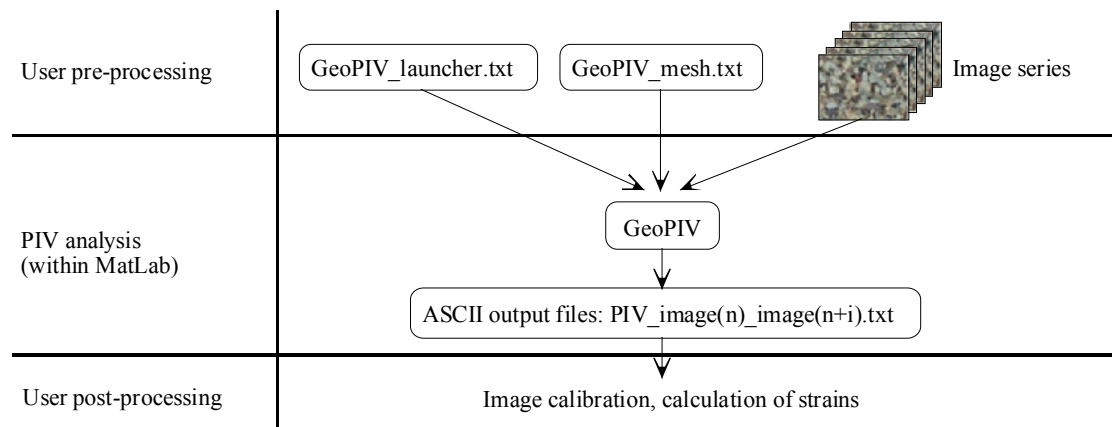
| User pre-processing | GeoPIV_launcher.txt | GeoPIV_mesh.txt | Image series |
|---|---|---|---|
| PIV analysis (within MatLab) | | GeoPIV | |
| | | ASCII output files: PIV_image(n)_image(n+i).txt | |
| User post-processing | | Image calibration, calculation of strains | |

Figure 4. GeoPIV software usage

## *3.1 Input file #1: GeoPIV7_launcher.txt*

The file GeoPIV7_launcher.txt is shown in Figure 5. This template contains the input variables for each PIV analysis. It is recommended that the name of this file be changed to identify each PIV run.

All lines preceded by the '%' symbol are ignored by GeoPIV, and can be used to store comments. The input variables are as follows:

GeoPIV7_mesh.txt    The ASCII input file containing the initial patch locations.

searchzonepixels    The value following this string is the largest displacement vector which GeoPIV will search for. This value is equal to half

the difference in size of the PIV patch and the search patch shown in Figure 1.

show_?            By switching the Boolean operator after these variables between 1 (on) and 0 (off), the display options during the PIV run can be changed. Figure 6. shows the displays activated by each variable. These displays allow the progress of the PIV run to be monitored, but add considerably to the calculation time.

spare_?           Spare. For future use.

C:\users\          This string indicates the location of the image files. By referring to a remote directory, a single set of image files can be stored at one location, and multiple PIV runs, all stored in different directories, can be conducted without having to make multiple copies of the images.

leapfrog          The integer following this string indicates how often image 1 is updated. If leapfrog = 1, GeoPIV compares images 1 and 2, then 2 and 3, then 3 and 4, etc… This leads to a low measurement precision over a long series of images (since the measurement errors are summed as a random walk), but reduces the chance of wild vectors since patches are easily identifiable after only one displacement step. If leapfrog is set higher, for example 3, GeoPIV compares images 1 and 2, 1 and 3, and 1 and 4. At this point the initial image is updated, before comparing images 4 and 5, 4 and 6 etc… To improve precision, the leapfrog flag should be set as high as possible, without creating an unacceptably high number of wild vectors.

subpixelmeth      This flag is superceded, and should be set to one.

%[Images]         The images to be analysed should be listed below this heading. Most image formats are accepted, including .jpg, .gif, and .tif.

Figure 5. GeoPIV7_launcher.txt



showmesh = 1
(the mesh of patches is displayed)

showpatch = 1
(each patch is displayed)

showvector = 1
(the magnitude of each calculated
displacement vector is displayed)

showquiver = 1
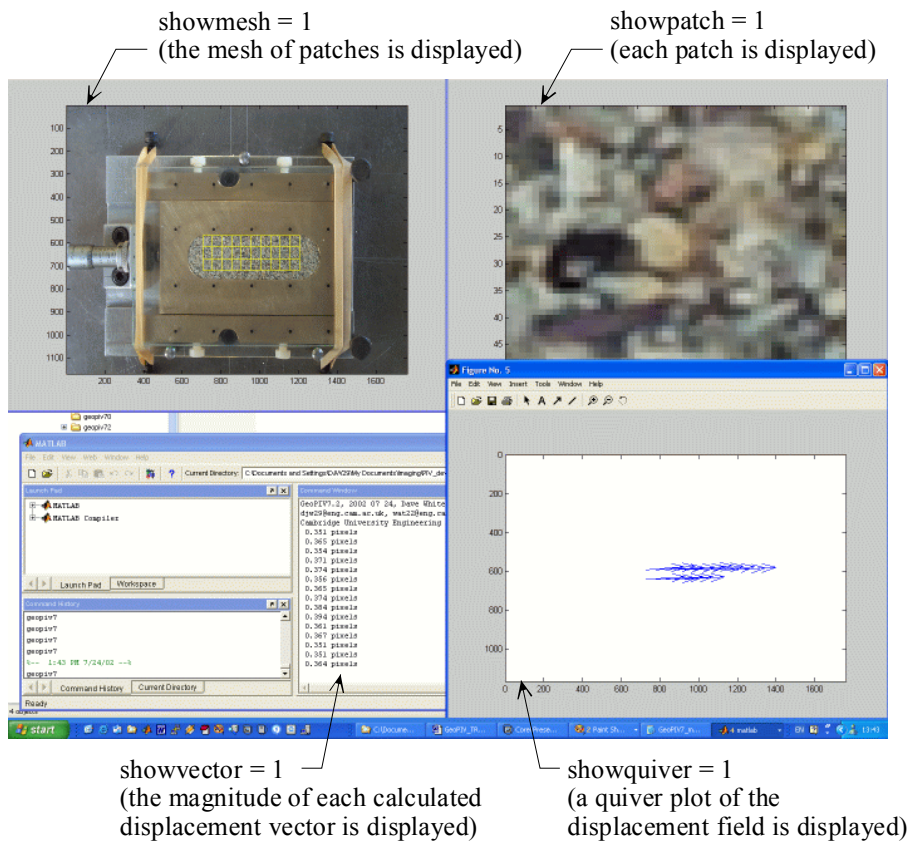(a quiver plot of the
displacement field is displayed)

Figure 6. Display options during GeoPIV analysis

## *3.2 Input file #2: GeoPIV7_mesh.txt*

The GeoPIV7_mesh.txt input file contains the locations of the initial mesh of PIV patches. This file is in ASCII format, but can be generated from a spreadsheet or using MatLab. The name of GeoPIV7_mesh.txt can be changed to allow easy identification of a particular mesh, with the change being passed through to the fifth line of GeoPIV7_launcher.txt.

Each row of GeoPIV7_mesh.txt defines a single patch. Each initial patch is identified by an ID number (column 1), the $(u,v)$ coordinates of its centre (columns 4 and 5), and its width, $L$, (column 8). The $(u,v)$ image coordinate system has its origin at the top left of the image, with $u$ increasing from left to right. A simple mesh file and the resulting grid of patches are shown in Figure 7.
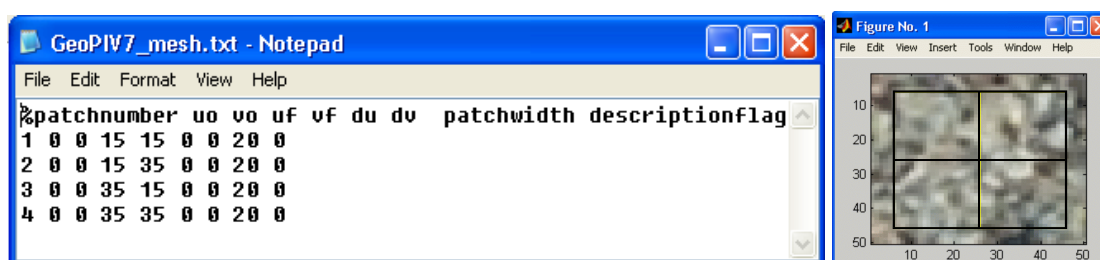


Figure 7. GeoPIV7_mesh.txt

## *3.3 Launching GeoPIV*

After preparing the two ASCII input files, the user launches GeoPIV from the MatLab command line, by typing GeoPIV7 (at the time of writing the latest release of GeoPIV is version 7). The following files must be on the MatLab path:

GeoPIV7.dll

load7.m

A pop-up box prompts the user to select the appropriate GeoPIV_launcher.txt file. All output files are created within the same directory as the selected launcher file.

After the launcher file is selected, the analysis begins. Any display options selected in the launcher file appear, and construction of the displacement field between the first

image pair starts. After each image pair has been analysed, an expected completion time is shown in the MatLab command window. Meanwhile, an ASCII output file is created after comparison of each image pair.

## 3.4 Output files: PIV_image(n)_image(n+i).txt

The ASCII output files have an identical format to the GeoPIV_mesh.txt files. Therefore, the output file from one PIV run can be used as the initial mesh for a subsequent analysis. Each output file has a filename composed of the two images being compared (Figure 8).

Each row of the output file corresponds to a single PIV patch. The first column indicates the patch ID number. The $2^{nd}$ and $3^{rd}$ columns indicate the coordinates of the patch in the first image. The $4^{th}$ and $5^{th}$ columns indicate the coordinates of the patch in the second image. The $6^{th}$ and $7^{th}$ columns show the corresponding displacement vector. Column 7 contains the patch width, $L$, carried over from the mesh file. For post-processing, it is usual to load the series of PIV output files created by a single run into a spreadsheet or MatLab.
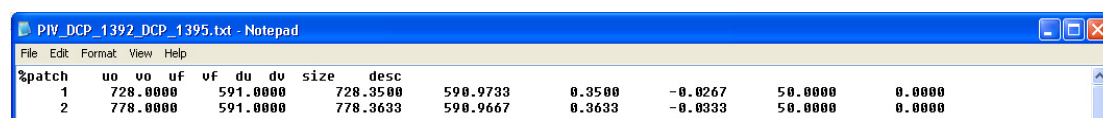


Figure 8. PIV output file

# 4 Troubleshooting

It should be noted that PIV analysis can be conducted badly, creating misleading or incorrect displacement data; the phrase "garbage in, garbage out" can be applied. The following section describes some of the pitfalls which can lead to invalid data. The Authors accept no responsibility for any data created using the GeoPIV software. Users should satisfy themselves that the data they have obtained is reliable.

Furthermore, good PIV analysis represents only one stage in the process of obtaining accurate and precise deformation data from a geotechnical test. As noted earlier, the accuracy of the resulting deformation data depends on the process used to convert image-space (PIV) measurements into object-space values.

## 4.1 Search zone set too small: wild vectors.

The search range over which GeoPIV searches for a displaced patch is set by the 'searchzonepixels' flag. This flag should be set higher than the largest expected displacement vector. If not, GeoPIV will not search far enough to locate the larger displacement vectors. Instead, wild vectors will be recorded. This problem can be surmounted by setting 'searchzonepixels' to be greater than the image width. In this case, GeoPIV will search the entire image, ensuring that each patch is located (assuming that it remains within the image). However, this approach will lead to an impractically long computation time. Therefore, a compromise is needed. The user should manually examine a typical image pair in order to estimate the displacement of the fastest moving point within the image. The 'searchzonepixels' flag should then be set comfortably above this value.

If a measured displacement vector is greater than 90% of the search range, an exclamation mark (!) will appear in the command window. This warns the user that the displacement field contains vectors that are approaching or are greater than the search range. The user may wish to rerun the analysis with a larger value of the 'searchzonepixels' flag.

## 4.2 Frame rate too low: wild vectors

If wild vectors continue to appear, even when searchzonepixels is set higher than the maximum expected displacement, the frame rate may be too low. This can result in excessive change in the appearance of each patch over each displacement step. This change in appearance may prevent correct identification of the patch. Correct identification is not possible if the correlation peak created when the initial and displaced patches overlay each other is drowned by the noise of the random correlation peaks created elsewhere on the correlation plane (Figure 1 shows a correlation plane in which the displaced patch position creates a single distinct peak).

This situation can be remedied by increasing the patch size. This reduces the influence of random changes in patch appearance. Alternatively, the experiment can be repeated with a higher frame rate, leading to less change in patch appearance between image pairs.

## *4.3 Patch size too large: strain field detail lost*

If large patches are used, the displacement field is 'smeared' within zones of localised deformation. Smaller patches produce improved spatial resolution of the displacement field. If a zone of localised deformation, for example a slip plane, is known to exist, and small patches cannot be used, it may be appropriate to establish an initial mesh consisting of lines of patches on either side of the localisation.

## *4.4 Patch size too small: wild vectors, reduced precision*

Smaller patches contain less information and are therefore more sensitive than large patches to changes in appearance due to distortion or unsteady lighting. This can lead to wild vectors. Also, small patches offer a lower measurement precision than large patches (Figure 3). These disadvantages are balanced by the improved spatial resolution of the displacement field.

## *4.5 Leapfrog flag set too low: reduced precision*

It should be noted that the values of measurement precision shown in Figure 3 are for a single small displacement step. If an series of $n$ images are analysed, with a leapfrog flag equal to $f$, the overall measurement error accumulated in the final image is equal to a random walk of length $\sqrt{(n-1)}$ divided by $f$. Therefore, the leapfrog flag should be set as high as possible, to maximise precision, notwithstanding the comments in Section 4.6.

## *4.6 Leapfrog set too high: wild vectors*

A high leapfrog flag can lead to wild vectors if a patch has become unrecognisable over the $f$ image steps between updating of the initial patch (cf. Section 4.2). Also, the cumulative displacement of the patch may exceed the search range (cf. Section 4.1 )

## *4.7 Scratched viewing window: 'stuck' patches*

Small scratches on the viewing window can cause image patches to become 'stuck'. This occurs if the stationary image content due to the scratch outweighs the moving content created by the soil. Larger patches that reach beyond the scratch may overcome this problem.

## *4.8 Insufficient texture: wild vectors*

If the image does not contain sufficient texture, i.e. there is a low spatial variation in brightness, the correlation peak created by the displaced patch may not exceed the random noise on the correlation plane. If the images under analysis contain zones of constant brightness, larger patches may be needed to straddle these zones and create an identifiable correlation peak.

# 5 Conclusions

GeoPIV is a MatLab module which implements Particle Image Velocimetry (PIV) in a manner suited to geotechnical testing. This report describes the practical details of using GeoPIV to measure displacement fields from digital images. In addition, some common pitfalls are described.

Also, the performance of the GeoPIV software is summarised, and the references from which further information can be found are listed. The software can be made available for research external to CUED; contact the Authors for further information.

# 6 Contact details

The Authors can be contacted as follows if required:

Dave White

djw29@eng.cam.ac.uk

Dept. of Engineering, Trumpington Street, Cambridge, CB2 1PZ

Andy Take

wat22@eng.cam.ac.uk

Dept. of Engineering, Trumpington Street, Cambridge, CB2 1PZ

Any comments on the performance or usability of this software would be appreciated.

# 7 References

Adrian R.J. 1991. Particle imaging techniques for experimental fluid mechanics. Annual review of fluid mechanics 23:261-304

Christensen K.T., Soloff S.M. & Adrian R.J. 2001. PIV Sleuth: integrated Particle Image Velocimetry (PIV) Interrogation/Validation Software. Tecehnical Report 943, Dept. of Theoretical & Applied Mechanics, Univ. Illinois at Urbana-Champaign.

Take W.A. 2002. The influence seasonal moisture cycles on clay slopes. University of Cambridge PhD Dissertation

Taylor R.N., Grant R.J., Robson S. & Kuwano J. 1998. An image analysis system for determining plane and 3-D displacements in soil models. Proceedings of Centrifuge '98, 73-78 pub. Balkema, Rotterdam.

White D.J., Take W.A, Bolton M.D. (2001a) Measuring soil deformation in geotechnical models using digital images and PIV analysis. Proceedings of the 10th International Conference on Computer Methods and Advances in Geomechanics. Tucson, Arizona. pp 997-1002 pub. Balkema, Rotterdam.

White D. J., Take W.A, Bolton M.D. & Munachen S.E. (2001b) A deformation measuring system for geotechnical testing based on digital imaging, close-range photogrammetry, and PIV image analysis. Proceedings of the 15th International Conference on Soil Mechanics and Geotechnical Engineering. Istanbul, Turkey. pp 539-542. pub. Balkema, Rotterdam.

White D. J. (2002) An investigation into the behaviour of pressed-in piles. University of Cambridge PhD Dissertation.