Correlation Imaging Velocimetry at the Coriolis facility

All comments or questions are welcome, contact sommeria@coriolis-legi.org

14th February 2003

Contents

1 General principle:

The method used at the Coriolis facility relies on pattern matching between two successive images of the flow, as described by Fincham and Spedding (1997) and Fincham and Delerce (2000). The displacement field proportional to velocity is obtained from the direct cross-correlation (covariance) between image pairs, limited to small boxes. The displacement vector on each box is determined as the location of the covariance(3.2.3) maximum. The flow is generally seeded with particles, but dye patterns can be used as well (see Pierini et al. 2002, DAO 644, 1-21). The seeding with particles(2.1) must be dense (typically 0.05 per image pixel) to get good statistics for correlations.

This technique has to be distinguished from individual particle tracking (PTV). Such tracking is practically possible only with a few hundred particles, generally leaving holes in the velocity measurement. Furthermore it requires sufficiently big particles, a few mm for the large fields of view used (up to 3 x 3 m^2), and they may not faithfully follow the flow due to their inertia. Of course such individual tracking is appropriate for Lagrangian data (we refer to L. Stig and T. Mc Climan at SINTEF, Norway, for a stereoscopic 3D particle tracking method in large scale facilities). The present correlation method is better for Eulerian fields, as long as a dense particle seeding is provided.

The field of view is illuminated with a laser(2.2) sheet, and we get the velocity field projection on this plane of measurement. By scanning twice a volume with this laser sheet, and comparing two images obtained at the same position, velocity is obtained in a volume, but still projected in the plane of the image. We have also developed a technique of full 3D correlation, but this will not be discussed here.

We discuss in section 2 the choice of the physical parameters, in order to obtain images of good quality, a primary condition for a successful processing. The available software for the complete CIV processing is discussed in section 3, and the precision of the results is discussed in section 5. A step by step procedure for processing is described in section 4. More detail documentation of the processing programmes (CIVx and uvmat) and file format are available at http://www.civproject.org/documentation.htm.

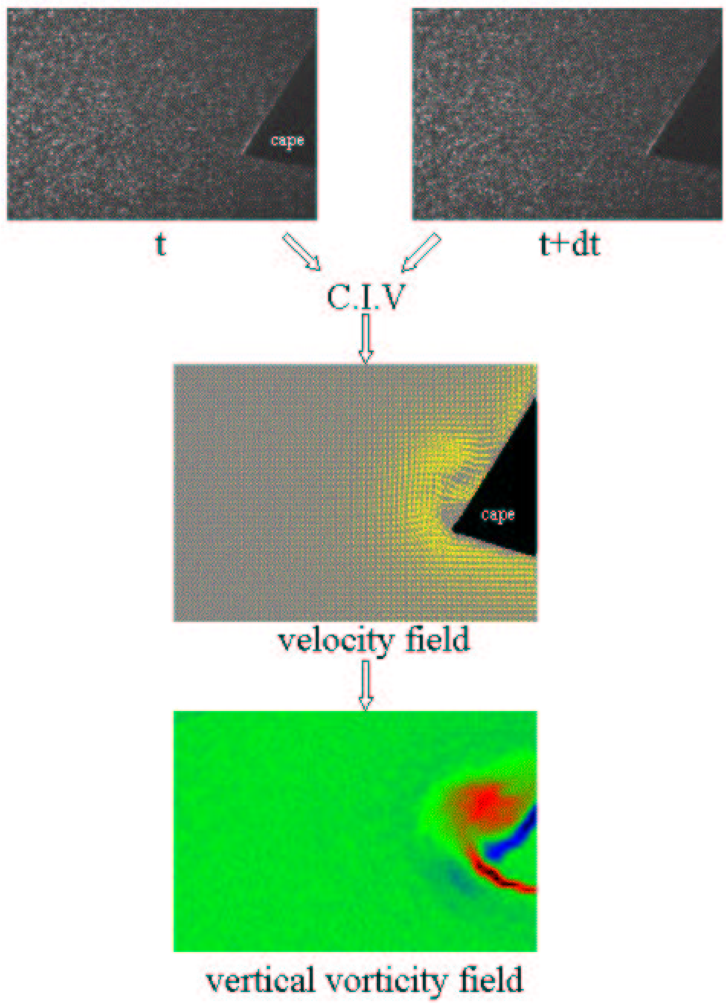

Figure 1: Correlation Image Velocimetry: the velocity field is obtained from the cross-correlation betweeen the particle images at time t and at time $t + dt$. The vorticity field, represented in color levels below, is deduced by differentiation.

2 Physical conditions:

2.1 Particles:

The particles must be sufficiently bright to provide contrasted patterns (not a milky image), but they must be small enough to minimise inertia and sedimentation effects, as well as laser beam absorbsion. To avoid sedimentation, the density matching with the fluid is critical. In practice we use two kinds of particles.

- Case of homogeneous water density:

In this case, we use pliolite particles (sold as an additive for painting by the company Goodyear). We select a particle size about 0.3 mm (the particles are grinded and sorted between sieves of mesh 0.25 mm and 0.40 mm). These particles have the advantage of being bright and available in large quantities with a well defined density, 1.0236 g/cm^3. Using salinity, we adjust the density of water to this value, in fact a little lower (1.023) as sedimentation in the bottom is preferable than accumulation at the free surface, which prevents visualisation. The particles sedimenting on the bottom (after a few hours) must be resuspended by moving a thin bar on the bottom. We can refill the upper part of the fluid layer by spraying with a perforated tube, slowly rotating above the surface. Both operations leave some residual turbulence, and a compromise must be found with particle rarefaction by sedimentation.

- Stratified fluid:

In stratified cases, we use expandable polystyrene particles of the same diameter (Gedexcel of the company HCC), which can be adjusted in density. These particles indeed swell by heating due to the formation of small pentane gas bubbles (this is used to make expended polystyrene). By heating in water at 70-80 ^{0}C, the density randomly decreases by a few percents from its initial value 1.05. After this heating, the particles are sorted by density in a vertical barrel filled with a stratified density. Particles with a given density range (3 10^{-3}g/cm^3) are stored in bottles (in a humid state). An equal quantity of particles from each density bin is introduced in a linearly stratified tank.

Note that in all cases, a tensio-active chemical, Ilfotol (used in photographic film processing), is mixed with water (concentration 2 10^{-4}) to favour the wetting of particles and avoid agglomeration.

A sufficient number of particles (typically 0.05 per pixel) is essential to achieve good CIV results. For a field 1000x1000 pixels, we need therefore 50 000 particles. If the corresponding area is 2.5 x 2.5 m^2 and the laser sheet is 1 cm thick, the density of particles is therefore about 1.25 particle/cm^3. The mass of each particle, with radius a=0.2 mm, is 3.3x 10^{-5} g, so the corresponding total particle mass for our full 100 m^3 tank is 4 kg (1.25 x 10^8 particles). The mass concentration is 3x 10^{-5}, which does not perturb the flow properties. Damping of laser light by scattering is however a serious drawback of excessive particle density. Taking a scattering cross section πa^2=0.0012 cm^2 for each particle, and the 100 particles found in a cm^2 beam section of length 100 cm have a total cross section 0.12 cm^2. The laser beam damping is therefore 12 % per meter, 40% reduction over the typical 4 m length of the laser sheet.

Note that the (maximum) power scattered by each particle is $\pi a^2 P(Wd)^{-1}$, where P is the total power in the laser sheet, with width W and thickness d=1 cm. For W=250 cm and P=5 Watts; we get 2.4 10^{-5}Watt. At a viewing distance 4 m, a lens 20 mm in diameter captures a solid angle 2 10^{-6} rd^2, so that, assuming isotropic scattering (which is far from reality), a power 0.4 10^{-10} Watts reaches the camera sensor. With 1/60 s exposure, this corresponds to 0.7 10^{-12} J = 5 10^6 eV=2 10^6 photons. Another factor to consider is contrast: the luminosity of the particle must be compared to the background luminosity of the area corresponding to one pixel.

To get higher spatial resolution, with smaller fields of view, much higher particle density would be needed, for instance 16 times more for a 0.6 x 0.6 m^2 field. Then smaller particles must be used, 4 times smaller in diameter, to avoid laser sheet absorbsion. The viewing distance would be also 4 times smaller, with the same scattering power received in the camera.

Extension to larger fields of view (5x5 m^2) with a higher resolution camera (2000x2000 pixels) would also require smaller particles (typically by a factor $\sqrt{2}$, to keep the same laser absorbsion over a twice longer length. As the viewing distance should be also increased (say by a factor $\sqrt{2}$), the captured scattered power drops by a factor of 4.

2.2 Laser illumination:

We use an Argon laser with power 8 Watts, with optical fibres and an optical system to generate the laser sheet. Double pulsed Yag lasers are classically used in PIV technique (for an application to a large hydraulic facility, the 100 m long wave channel of INHA at Barcelona, see http://lim-ciircupc.es/eng/CIEM/index.htm). They provide an intense peak power on a very short time, so that fast velocities can be captured. However, the repetition rate is small (typically 20 Hz), optical fibres would be damaged by excessive pulse power, synchronisation with camera is delicate, and a careful alignment of the two laser beams is needed.

Practical operations are thus much easier with a steady laser, and the flow evolution can be resolved in time, or the 3D structure obtained by rapid scanning of the laser sheet. The steady laser is therefore quite appropriate for our flows with moderate velocity, up to typically 10-20 cm/s.

We produce the laser sheet with an oscillating mirror (model 6850 of Cambridge Technology), at frequency 120 Hz, synchronized with the images by the electronic pulses emitted by the camera. The total angular spread in water can reach 50^0, but it can be adjusted to optimize luminosity in a given field of view. Producing such a wide and uniform laser sheet by scanning is another advantage of operating with a steady laser.

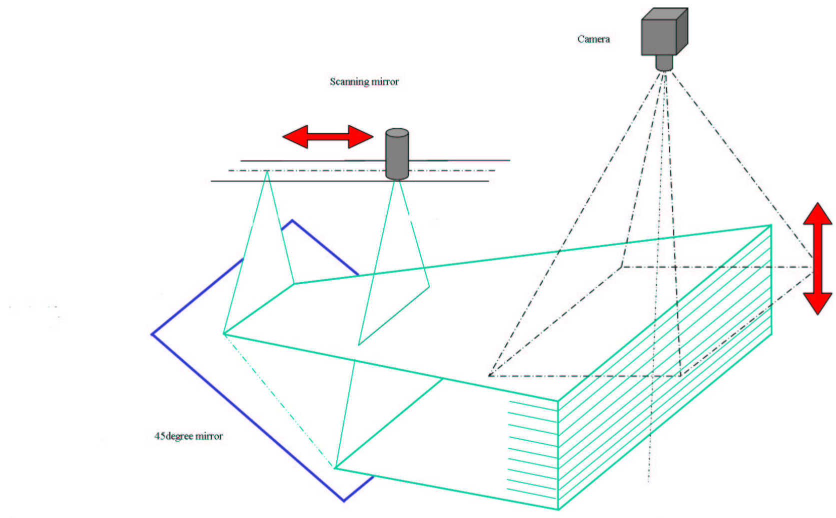

Figure 2: laser sheet scanning for CIV in a volume.

2.3 Camera:

The quality of the camera is a key condition for good CIV measurements: high sensitivity, low noise, high spatial resolution are desirable. A high image rate is needed for 3D scanning.

An analogic camera can be used, with a numerical image acquisition board in the computer, provided it has a progressive scan schutter (it makes each image in a single scan, while the usual video format is made of two interleaved scans, separated by half the frame period). Cameras with numerical output provide better results. We use two kinds, the Pulnix TM-9701 and the more recent SMD 1M60.

The Pulnix has resolution of 768*484 pixels, with 256 grey levels (8 bits), image rate 30 per second, sensitivity : 10 microV/e- (which is about 1 lux with f=1.4). Cell size is 11.6*13.6 μm^2, imager is 2/3"progressive scan interline transfer CCD. The image acquisition board is from Imaging Technology , PC-DIG model with 2Mo memory.

The SMD 1M60 digital camera provides high-sensitivity 12 bit images with 1024*1024 spatial resolution at up to 60 frames per second. The sensitivity is 8 microV/e- , pixel size is 14μm*14μm and the fill factor is about 70%.

The output is split toward two Imaging Technology framegrabber to avoid the PCI bus limit at 133Mo/s; hence the computer has two PCI 32bits/33MHz bus and the signal split is done with a home made "Y" cable. A commercial system based on the same concept is no available, the PIXCI D3X (http://www.epixinc.com/products/pixci_d3x.h

Both systems use the ITEX libraries, which provides a set of tools for controlling the camera.

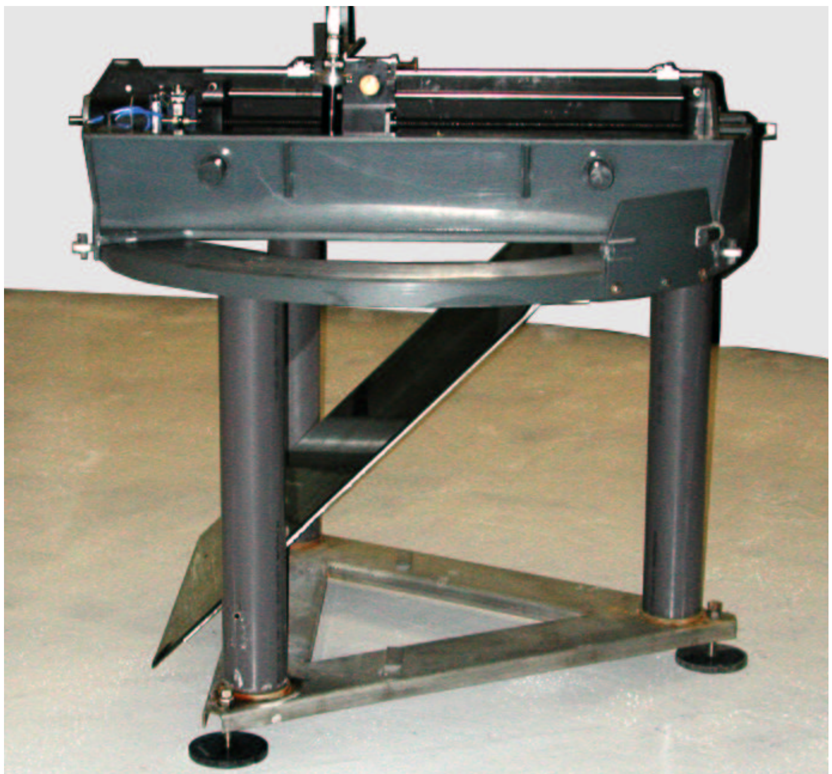

Figure 3: system of translation for the laser sheet. The laser beam arrives from the top by optical fibers on a small oscillating mirror (see closer view in fig. 4) which generates a vertical laser sheet, which is deviated into a horizontal sheet by the 45^0 mirror. A laser sheet in the vertical plane can be generated as well by rotating the whole system by 90^0 with respect to the mirror.

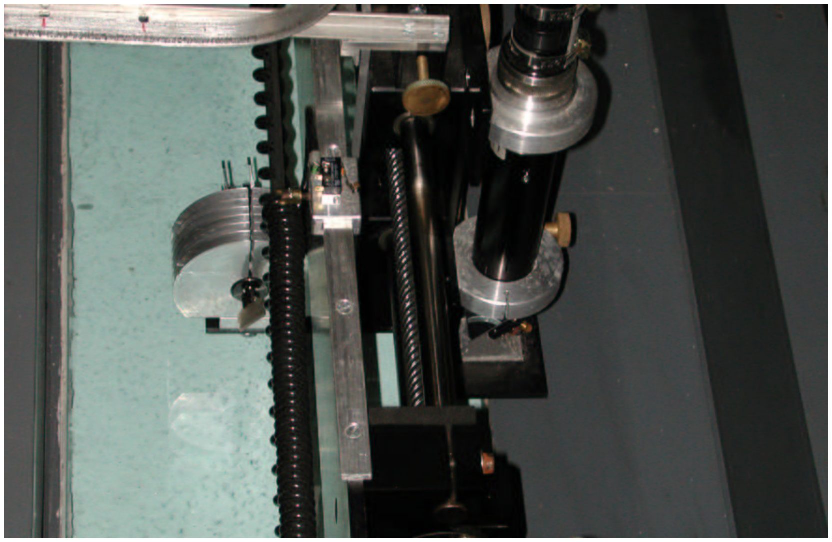

Figure 4: closer view of the laser sheet generator, seen from the top.

	PulnixTM9701	SMD/DALSA 1M60
Frequency	30 Hz	60 Hz (1024x1024) 110 Hz (512 x512)
Resolution	768 x 484	1024x1024 pixels
Digitalisation	8 bits (256 grey levels)	12 bits (4096 grey levels)
Sensitivity	10 microV/e-	8 microV/e-
Cell size	11.6μm*13.6 μm	14μm*14μm
CCD size	9.6 x6.7 mm2	14x14 mm2
frame grabber	PC-DIG CORECO IMAGING	PC-DIG CORECO IMAGING

2.4 Objective lens:

We need a lens with high luminosity, characterized by the aperture $u = f/y$, the ratio of the focal length f to the useful lens diameter y. The light energy received at a distance D from a particle is proportional to the solid angle $(y/D)^2$ captured by the lens, see sketch. Decreasing the distance D is favourable, but there is an optimum, as y is limited by lens aperture and the magnification condition $l/f = L/D$ (see sketch in fig. 5 right), where l is the sensor length and L the corresponding length on the field of view (see sketch in fig. 6 left). We make the approximation that the image forms at a distance f from the lens (which is strictly true only for an object at infinity). Therefore

$(y/D)^2 = (l/L)^2 u^{-2}$

which depends only on the objective aperture u, for a given field of view and sensor.

For a given aperture, it is advantageous to use a large distance D, providing less geometric distortion and a less critical focussing. This is important for

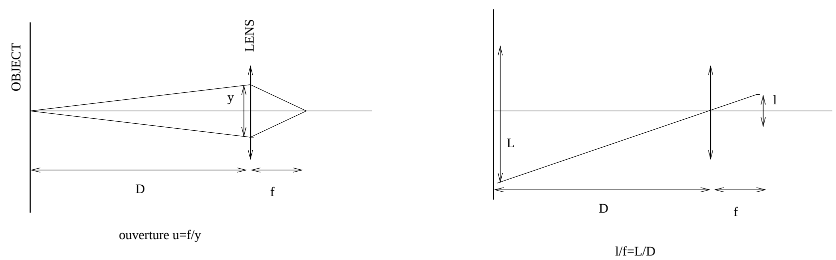

Figure 5: magnification(right) and aperture (left)

measurement in a volume, since the distance to the camera is changing during the laser sheet scanning. Note that a perfect focussing is not necessary advantageous. Some fuzziness of the particle images indeed results in wider and smoother correlation curves, which could lead to a better subpixel precision (although we did not really test that).

We have plotted in fig. 6the visible field length L for our different lenses, with the two cameras, at viewing distances 3 m. For other distances, the field varies of course in proportion to the distance. The Pulnix(2.3) camera has a sensor 6.7x9.6 mm, while the SMD(2.3) has a sensor 14x14 mm, so the field is larger with the later. Objectives with very large aperture (u=0.9) are available only with diameter 1 inch, which is designed for a sensor up to 16 mm in diagonal. This is all right for the Pulnix, but too small for the SMD2.3), for which the corners of the field are darkened. However this drawback is largely compensated by the higher luminosity for most part of the image. Objectives for 24x36 photography (Sigma, Olympus or Nikon) provide perfect image quality, but with lower luminosity.

Our best choice for large fields of view with the SMD2.3) camera is the Schneider 25 mm, which provides a field 2.5 x 2.5 m2 at distance 4 m. For smaller fields, the Nikon 35 mm or Olympus 50 mm provide images of excellent quality with still a good luminosity (aperture 1.4).

The deformation effects are small in air, except for the largest fields of view, i.e. the smallest focal length, 17 mm or below. When we look from air inside water, deformation is due to refraction at the interface. A programme is available in the CIVproject package (see3) to account for this deformation, by relating the (x, y) coordinates on the image to the physical (x, y) positions, depending on the lens and water depth below the free surface (assumed plane).

2.5 Frame period and contrast:

The camera parameters are directly set from Programs/SMDIMAP/SMDCAM accessible in the "Start" menu of "Windows" (also accessible in Civit). This is the frame period in s (close to the exposure time of an image), the offset and electronic gain (1 or 4).

Longer frame periods (exposure times) result in better image luminosity.

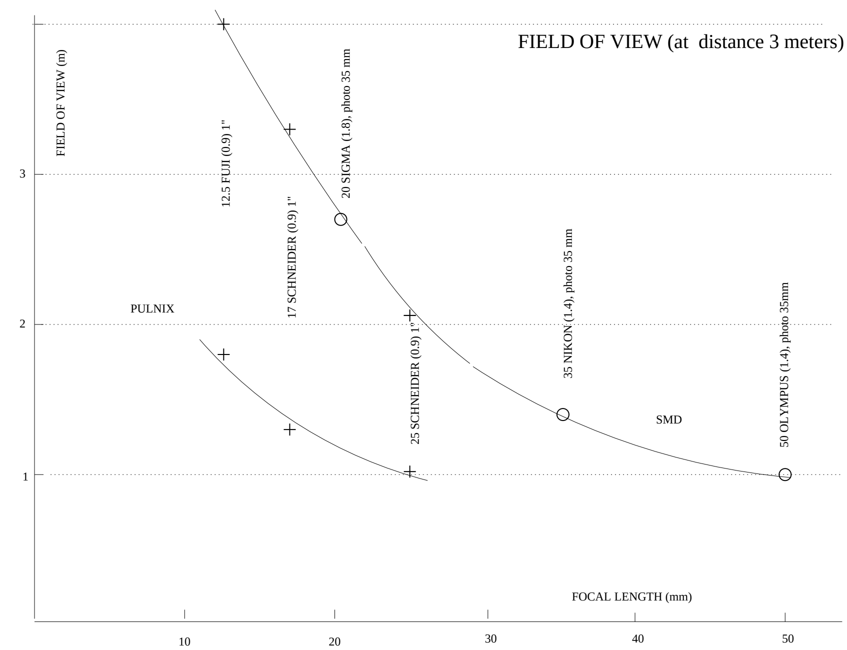

Figure 6:
Field of view at 3 meter distance with different available lenses

The particle displacement during the exposure time leads to a wider correlation curve which may by itself improve the measurement. The exposure time must be however sufficiently short to avoid streak effects, which would lead to excessively flat correlation curves, whose maximum is not well defined. A good compromize would be to choose an exposure time for which the maximum displacement is about one pixel.

Then the luminosity and contrast (related to offset and gain) are to be chosen to avoid saturation for both minimum (no point with zero value) and maximum luminosity (no point with luminosity 256 for the Pulnix(2.3) or 4096 for the SMD(2.3). Check the image histogram (visible on the Civit screen). Do not use the option "antibluming" for the SMD at low luminosity: it produces spurious vertical lines.

Of course the background must be black, and no significant reflection from the free surface must be visible. Otherwise it will result in spurious correlation peaks in CIV algorithms, associated with the velociy of the background (generally zero). Adjustment of contrast and luminosity may help suppressing these effects. Note that there is a possibility of introducing a mask(3.2.4) in the processing, to remove edges in which the image would be polluted.

An example of excellent image is given in fig. 1. Note the grey scale "texture" of the image, in which individual particles cannot be really distinguished. Sharp contrasts (like black and white images) are not favourable, as well as images with

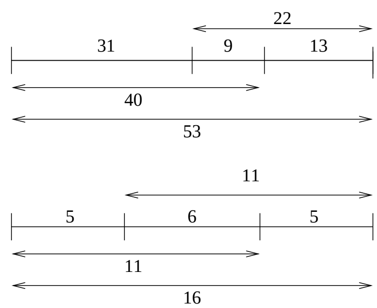

Figure 7: example of time intervals in a burst of 4 images, expressed in units of 1/15 s, a) allows a large range of intervals, b) allows to compare two intervals centered at the same time.

rare isolated particles.

2.6 Bursts and time lapses:

The time lapse between two images must be sufficiently short so that the displacement remains proportional to the local velocity, and correlations are not lost by deformation(3.5). Furthermore a majority of the particles must not move in and out of the laser sheet by their transverse velocity. Shorter time lapse leads to a larger relative error in the measurement of displacement. A typical displacement of 5 to 10 pixels is generally optimum for a first CIV processing, and possibly increased with hierarchical iterations(3.5).

To be able to optimize the time lapses, we record "bursts" of 3 or 4 images (expressed in units of the frame period in Civit2.7). We can use any pair of images chosen in this set in the CIV software, so that during data processing, the time lapse can be chosen (with four images) among t_1, t_2 ,t_3, or the sums $t_1 + t_2$, $t_2 + t_3$, $t_1 + t_2 + t_3$.

For instance, for a field 2.5x2.5 $m^2$with typical velocity 2.5 cm/s, one pixel (2.5 mm) is travelled in 0.1 s, so we recommand a frame period of 1/15 s. To allow an optimum choice of time intervals, it is good to use a burst with three or four images, as sketched in fig. 7. This allows to adjust the time interval for image pairs. If the maximum range of intervals is wanted (e.g. if the velocity values strongly depend on position or time), the choice of fig7a is appropriate. The choice of fig. 7b allows less range, but it is optimum to estimate the precision by comparing the velocity field from the middle interval (with 6/15 s) to the field averaged from the two side intervals (with 5/15 s), and the field obtained with the total interval 16/15 s (see **??**).

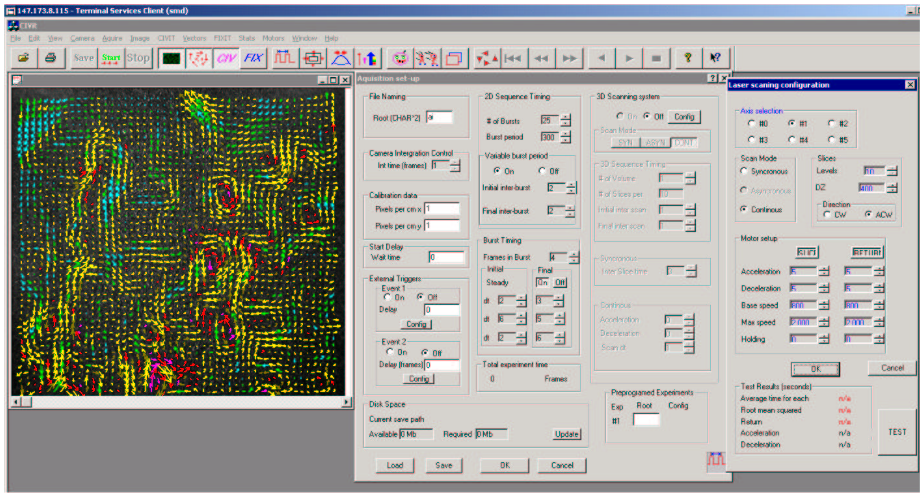

Figure 8: Main menu of Civit

2.7 The menu of Civit:

The main menu of Civit is reproduced in fig. 8. The acquisition parameters are set in Aquire/Configure. This menu, shown also in fig. 8 first allows to choose the path (directory) and file Naming(3.6) of the image series (with 2 characters).

The "Camera Integration" gives the frame period (in multiple of the elementary one, 1/60 s for the SMD(2.3) and 1/30 s for the Pulnix(2.3)). This provides the unit used for all the other times. The "calibration data" gives the scale in pixels/cm (it has to be introduced manually) . We can set also the Wait time before starting the image acquisition (no external trigger is presently available).

In the second column, one sets the image acquisition timing. We first set the total number of bursts and their period (i. e. the sampling period for the velocity evolution). This period can be made variable in time, which can be useful for an optimal sampling of a decaying flow. The Burst Timing sets the number of images in a burst and the successive intervals t_1, t_2, t_3 (always expressed in frame periods in the intervals between images, add one frame to get the time intervals). These intervals can be made variable in time with the option "variable burst period".

The parameters for the laser sheet scanning can be also set in the third column of the menu.

NOTE: the parameters of an experiment are put in a file.exp. Don't forget to save this file, using the corresponding option in the Civit menu.

2.8 CIV in multi-planes and volumes:

To get the flow structure in a volume, we have two possibilities.

- Multiplane 2D CIV. The laser sheet position moves at successive positions with regular spacing, and one burst is made at each position. It takes about one second to change position, and about the same to make a burst. Therefore, the total time for scanning is long, about 10 s, depending on the number of levels, and the flow can evolve during that time. The advantage is however that is no practical limitation to the measured velocity (it is like in 2D CIV).

- CIV in volume: in this case the laser sheet is sweeping the volume a first time, come back to its origin, and then sweep it again. Correlation is made between two corresponding images at the same position. In the present state of the technique, the minimum time interval t for the correlation is 2s, so that the method is limited to flows with moderate velocity, less than about 1 cm/s. In contrast the scanning time is short (less than 0.5 s), so we really get an instant scanning of the velocity field, unlike in the multiplane case.

3 Processing software:

3.1 Available software:

- The image acquisition, and the laser sheet displacement, are controlled by the software CivIT. It also performs and visualizes a simple CIV processing to check in real time the quality of the images. CivIT, written is C language, is commercially available from Fincham and Spedding.

- The CIV processing can be performed using the VSV interface, also commercially available from Fincham and Spedding. It contains a graphic interface written with the langage PVwave, also used to compute and visualize various flow data (e.g. divergence and vorticity). The CIV algorithms, called by the VSV interface, are in Fortran. The compiled (executable) version is in a directory CIVbin.

- The new algorithms described in Fincham&Delerce(2000) are now open source and available on the web site http://www.civproject.org . A web interface, CIVx, is also available which permits the computation distribution over a network of standard PC. The code, written in Fortran 90, is vectorized. It is deposited as a General Public Licence, and the sources can be downloaded from http://www.jarunee.org/civproject. The Web browser interface is written in the langage perl_cgi, so it can be used in a variety of computer systems (in particular Windows and Linux). At the Coriolis facility, it can be accessed with any Web browser at the address http://Z4/CIV. Online help is available. See also http://www.civproject.org/documentation.htm for various documentation.

- A user interface for image analysis and post-processing in MATLAB is also available. At the Coriolis facility, it is in the directory \\C-RED\civbin3\uvmat.

To install it copy uvmat.fig (the graphic interface), uvmat.m (the associated functions) and uvmat_doc.html (the online help) in the current MATLAB working directory. Once it is installed, just type the command uvmat under the MATLAB prompt.

3.2 Basic algorithm

3.2.1 pattern-box:

We first need to set the elementary box sizes B_x and B_y (in pixels) in which correlations will be calculated along each direction x and y. To estimate the minimum value required for these parameters, open an image (see section 3.6), and look at the minimum size of a square which always contains at least 5 particles. In practice a value of B_x and B_y between 21 and 31 is chosen. Larger values lead to lower spatial resolution: the velocity obtained is an average in this elementary box. In addition, the actual motion is a pure translation only in a local limit: deformation effect increases in proportion with the box size, leading to deterioration in the quality of the correlation (see section 3.5 for a discussion of this effect).

For a seeding with particles at a uniform mean density σ (in particles per pixel), the probability of finding n point particles in a rectangle B_xB_y is the Poisson distribution $P_\mu(n) = exp(-\mu n)/n!$, where μ is the mean particle number $\sigma B_x B_y$ in the square. For σ=0.05 and $B_x = B_y$=21, we have μ=20. Then the Poisson distribution gives a total probability 1.6 $10^{-5}$to get n< 5 particles. For a twice lower density, μ=10, this probability rises to 2.9 %, resulting in a significant number of false vectors(3.3) (even for a uniform seeding). We need then to use a larger box, or increase the particle density. Choosing different values of B_x and B_y is justified for rectangular pixels (e.g. with the Pulnix camera) or in the presence of a strong shear. For instance in a boundary layer, we can reduce the size transverse to the wall and increase the streamwise size.

3.2.2 search-box:

The search box must contain the expected optimum pattern position on the second image. It is defined by its centre, equal to zero in the absence of a priori knowledge, and its size in each direction S_x and S_y. An excessive value of this range does not change the result, but it increases the computation cost, and may lead to the appearance of false vectors(3.3) due to spurious covariance(3.2.3) maxima. If the search range is too small, the true maximum covariance(3.2.3) is missed and false vectors(3.3) are obtained . If the maximum is right on the edge of the search zone, the CIVx programme yields a flag vec_F=-2 (and puts the maximum covariance to 0).

After completion of CIV, you must check on the histogram of displacements along each coordinate that you did not cut large displacement values (the histogram must drop to 0 for the maximum accessible displacements $\pm[(S_x - B_x)/2 - 1]$ or $\pm[(S_y - B_y)/2 - 1]$. This histogram is shown by uvmat .

3.2.3 The cross-correlation (covariance) function:

We work with the pixel intensities $I_a(k,l)$ and $I_b(k+i,l+j)$, $k = 1..B_x, l = 1...B_y$, in a rectangular box of size B_x and B_y. The subscripts a and b refer to the two images between which the cross-correlation is calculated. The index origin (1,1) is set at the lower left corner of the box of the first image. The second box is shifted from the first one by the vector (i,j).

We first calculate the box intensity averages

$$\overline{I_a} = \frac{1}{B_x B_y} \sum_{k=1}^{B_x} \sum_{l=1}^{B_y} I_a(k,l)$$

$$\overline{I_b} = \frac{1}{B_x B_y} \sum_{k=1}^{B_x} \sum_{l=1}^{B_y} I_b(k+i,l+j)$$

We substract this average to each intensity, and calculate the cross-correlation normalized by the variance, or covariance $c(i,j)$ as

$$c(i,j) = \frac{\sum_{k=1}^{B_x} \sum_{l=1}^{B_y} (I_a(k,l) - \overline{I_a})(I_b(k+i,l+j) - \overline{I_b})}{[\sum_{k=1}^{B_x} \sum_{l=1}^{B_y} (I_a(k,l) - \overline{I_a})^2 \ \sum_{k=1}^{B_x} \sum_{l=1}^{B_y} (I_b(k+i,l+j) - \overline{I_b})^2]^{1/2}}$$

for each displacement (i,j) allowed by the search box(3.2.2). The correlation is often calculated by Fourier transforms in PIV software. The direct computation used here is more demanding in computing cost, but it allows more flexibility and precision for small search boxes, and it avoids spurious effects of Fourier transforms.

The velocity vector is obtained as the displacement which maximizes this covariance, multiplied by the geometrical scale and time interval dt. A key feature of CIV is the interpolation of this covariance to non integer displacements in order to reach sub-pixel precision. A thin-plate 2D interpolation method has been chosen here. It compares favorably with other interpolation methods in some test cases studied by Fincham and Spedding (1997). The resulting covariance function is piece-wise polynomial (of cubic order), so that its maximum can be precisely obtained. It minimizes a linear combination of global curvature and distance to the integer values $c(i,j)$. This is therefore not a pure interpolation, but some smoothing is introduced, depending on a parameter ρ.

With any interpolation method, there is some systematic bias of the covariance maximum with respect to integer values, on which the covariance has been initially defined. This deffect, called peak-locking(5.3), is one of the main source of uncertainty in CIV.

To minimize this peak-locking effect, it is favorable to have wide cross-correlation peak, for instance with large apparent particle sizes. Note that a high particle density leads to apparent clustering, hence larger cross-correlation peak. Indeed for each particle, we have a probability about 8σ to have another particle in one of the eight neighbour cells. This pair has itself a probability 10σ to be connected to a third particle. For σ=0.05, these probabilities are close to one half, so that a significant proportion of the particles are in clusters, giving the visual impression of "textures" seen in fig. 1. A drop by a factor of 2 in particle density sharply decreases this apparent clustering effect, leading to more narrow covariance functions.

3.2.4 Measurement grid:

The cross-correlation can be calculated with the pattern-box(3.2.1) centered at any point on the first image (a), yielding a corresponding velocity vector. These points are located on a given measurement grid, which is either regularly spaced (default option), or defined by a grid file of x and y coordinates. The velocity vectors correspond to the position at the middle between the initial pattern-box center and the optimum displaced one. For each point (x, y) on the measurement grid(3.2.4), this point has coordinates $x + u/2, y + v/2$ where (u, v) is the velocity vector expressed in pixel displacement. The velocity field is therefore never obtained on a regular grid. The velocity field can be obtained on a regular grid by interpolation(6.1), which also provides the spatial derivatives .

A typical mesh for this measurement grid is $B_x/2$ (for instance along x), so that adjacent correlation subboxs overlap by one half. A smaller mesh (e.g. $B_x/4$) leads to a 3/4 overlap. The resulting redundancy is useful to reduce the errors and evaluate them by continuity(**??**). A velocity field on typically a 100x100 grid is then obtained.

Masks can be introduced to restrict the measurement domain. The goal is to restrict the processing to an illuminated sector, to mask the field beyond walls, or remove bad parts of the image. The mask is a 8bit image file (.raw or .png) with the same dimensions as the image to mask. A greyscale code is used: for an intensity $<$20, the velocity is set to 0 (for instance at a wall); for 20$<$intensity$<$200, the vector is undefined (but it will be determined by interpolation from neighbors in the patch(3.4) programme; for 200$<$intensity, the vector will be computed. Center of the pattern box?VERIFIER

3.3 False vectors:

A major difficulty in CIV is the occurence of secondary maxima of the covariance, which can spuriously exceed the maximum associated with the local displacement. One method for limitating such false velocity vectors is to restrict the search box, using some a priori knowledge of the velocity field. This knowledge is generally provided by a first CIV treatement, which can be thus refined in an iterative, hierarchical(3.5) way. The a priori knowledge could be alternatively provided by a prediction of a numerical model with data assimilation of the previous flow evolution.

For the first CIV iteration, the option HART is recommanded. It is a good way to reduce the number of such false vectors, as described by D.P. Hart, "PIV error correction", Experiments in Fluids 29 (2000) 13): the covariance function at position (n, m) is multiplied by the covariance function at a neighborhing position (shifted by $B_x/2$, which was shown to be optimum). Since the maximum of the covariance(3.2.3) corresponding to the physical displacement should be close at neighborhing positions, the multiplication therefore enhances (but blurs) this true correlation peak, while secondary maxima due to noise are generally uncorrelated. This "correlation of correlation" is therefore smoother, with less false vectors (see fig. 9). Once the search zone is thus restricted, the programme comes back to the usual covariance(3.2.3) to determine the displacement with precision.

Remaining false vectors can be removed interactively using the uvmat programme. A threshold on the correlation value can be imposed, and flags on the correlation quality can be used (lack of convergence for the determination of the maximum, maximum at the edge of the search zone). Comparison between the raw civ results and its smoothed interpolation can be used, as well as comparison with another processing result with a shorter time interval. Other criteria are used by the fix programme (not yet documented). Finally hand removal can be used.

Note that when CIV2 is used (3.5), the elimination of vectors from CIV1 is only important in providing a good first guess for CIV2. The new results replace this first guess by an actual measurement from correlation maxima, on a restricted search zone.

3.4 Velocity interpolation (patch):

We are often interested in the velocity at any point, and different interpolation methods can be used for that purpose. The patch programme in CIVx provides interpolation on a regular grid, using the thin shell spline (ref...). It does not exactly interpolate, but it also smooths, depending on a parameter ρ. Such smoothing can improve the end result, by removing some noise, but it must be manipulated with care to avoid cutting off the smaller scales of motion. The spatial derivatives are also obtained from the spline results in this patch programme.

3.5 Hierarchical algorithm

The results from the first CIV processing (CIV1) can be improved in an iterative way by a succession of false vector removal (fix), interpolation to make a new guess, and CIV processing with reduced search zone, using the estimate from the previous processing. By this method convergence to local correlation maxima can be obtained, with displacements mutually consistent with their neighborhood.

Furthermore, in the second CIV processing (CIV2), the prior knowledge of the local deformation rate and rotation is used to refine the pattern matching, see fig.10. During the time lapse t, the local displacement differs from the pure translation by $sB_xt/2$ (or $sB_yt/2$) at the edge of the pattern-box. Deformation will deeply affect CIV when this shift becomes of the same order as the apparent

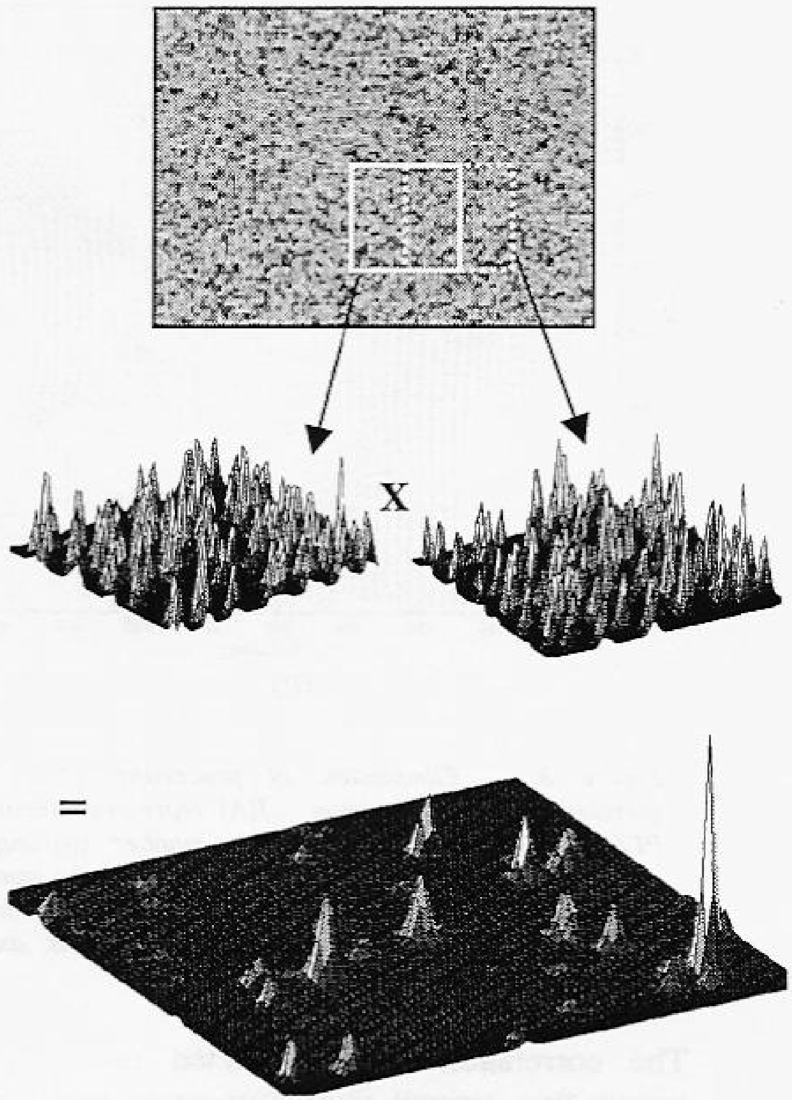

Figure 1. – *Elimination of correlation anomalies by multiplying the correlation tables from adjacent regions. Correlation values that do not appear in both tables are eliminated allowing tracer particle displacement to be resolved.*

Figure 9:

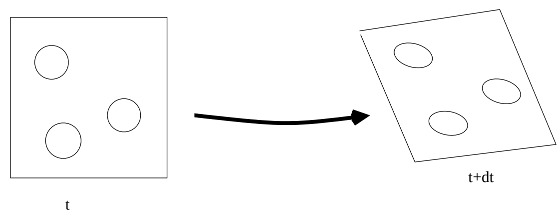

Figure 10:

particle size δ, or image correlation length. Estimating s from a typical velocity U as $s \sim U/L$, we get that deformation effects are important for$B_x d/2L > \delta$. Since $\delta \sim 1$ pixel, with a displacement $d = 10$ pixels, this corresponds to $B_x > 0.2L$.

To describe such deformation, it is necessary to interpolate the image itself between the integer pixel values. This is done by a spline interpolation of the image intensity in the pattern-box.

Taking into account this deformation effect allows, as well as prior information, allows to increase the time interval, and therefore the precision. The use of prior information may also allow to reduce the pattern-box and increase the spatial resolution.

3.6 Data format

A standardisation of the data format has been recently performed, allowing compatibility with other applications. It is described at http://www.civproject.org. The raw images are stored in the compressed (reversible) form '.png' , which saves a factor about 3 in memory, and can be read directly with standard image software. The output fields are stored in the 'netcdf' format which can be read directly by all computers (without problems with number coding), and which provides a clear information of the relevant parameters, allowing to track back the processing steps.

In the "acquire" menu of Civit(2.6), once the generic name (for instance "aa") has been set , a directory "aa" is created, including the successive image files aa001a, aa001b, aa001c,aa001d, for burst number 1, and similarly for the successive bursts.

The experimental parameters are recorded in a text file aa.civ with the following format.

278 ....number of bursts
1024 1024..........................pixel number in x and y
4 ....number of images in a burst
1 ....not relevant
0.066667 ....frame period of the camera
4.080000 4.080000.........scale in pixels/cm along x and y
4.080000 4.080000.........same
1 ....1 indicates 2D CIV

1 0.000000 7 17 56 1
2 10.666694 7 17 56 1
3 21.333498 7 17 56 1 .... burst number, time elpased from the beginning (in s), successive time intervals $t_1 t_2 t_3$ in frame periods.

Each image file can be read with the Matlab programme (6) (for the old, uncompressed image files) .P

4 CIV processing step by step:

4.1 Check image quality:

This can be done with the uvmat programme, with the field selection image. Movies can be made to check a whole sequence. By simultaneously selecting a pair of images, for instance a and b, we can visually appreciate the feeling of continuity of particle patterns between the two images. Use the zoom option to get a closer view.

A very dense seeding of particles is the primary condition to check. Check also the luminosity and contrast of the images: fully black or saturated parts of the images must be avoided (by adjusting the luminosity and contrast of the camera). Look at the image histogram to check that.

A common source of difficulty is the occurence of fixed patterns due to illumination of the background. This effect can lead to zero velocity vectors. To avoid this effect, the first condition is to improve the quality of the illumination, using screens to avoid spurious reflexions. A poor water clarity can also contribute to illuminate the background by diffusion of the laser light. As a last recourse, one can substract the average to a sequence of images (e.g. by the option field_average in uvmat. This substraction removes the fixed background (but as a tradoff adds some noise).

Don't forget to check the calibration for distances: make images with a reference grid in the view field with the same camera position and objective lens. Then the calibration pixel to cm can be obtained. In uvmat, the pixel index of any point can be obtained with the mouse: first check the zoom option (left click to zoom and right click to unzoom), then press the select pushbutton to get the mouse cursor, and left click on the mouse. The pixel indices in x and y (as well as the image luminosity) appear in the text window at the upper right.

Adjust the angle of the camera such that walls are parallel to the x or y coordinates: this makes further processing much easier. A mask can be made to exclude some regions of the images in the processing, for instance to take into account the domain boundaries. The matlab programme makemask can be used for that purpose.

4.2 Run CIV1:

Open the CIVx web interface at address http://Z4/CIV. Enter the file .civ in the directory of the image series to process. Enter possibly a mask file and a custom measurement grid, then click on processing options.

A new web page appears, where the processing parameters can be chosen. Use the default algorithm selection CIV1. Keep the default values for the algorithm option and FIX option on the right.

The image organisation options give the first and last image numbers to process, as well as an interval increment N step.

Choose the pattern box sizes (Ibx and Iby on the interface). The default values $Ibx = Iby = 21$ are generally appropriate for a first run with good images. Reduce it to optimize the spatial resolution, but then false vectors may appeart in zones with fiew particles. Increase the box size with poor images with few particles. To analyse a boundary layer, you can use a streamwise elongated box with the same area, for instance 11x41 or 5x81. See also 3.2.1.

The search box size (Isx and Isy) have to be chosen to contain the maximum pixel displacement in the field, see 3.2.2. Avoid excessive search box size, which favors the appearance of false vectors, and increases the computing time. If a systematic velocity direction is expected, you should introduce a systematic shift (ShiftX and ShiftY)), and reduce the search box accordingly (by ShiftX/2 and ShiftY/2 along each coordinate). If you have not introduced a grid file, choose a mesh for the grid of measurement points, in pixels.

The smoothing parameter ρ tunes the smoothing of the fit of the image correlation needed to find its maximum. The optimum choice should depend on the particle size and level of image noise. We do not have a clear criterium of choice, so just keep the default value 1.

Select the image pair chosen for the correlations.

Select the option log.file to get a record file of the processing events. If this log.file does not appear quickly in the image directory, look at http://z4/civ/civbot/TODO.txt whether your job is on the waiting list.

4.2 Check CIV1 results:

Analyse the civ1 results with the uvmat programme (field velocity).

The correlation values are indicated by colors red, green, blue (with thresholds adjustable by sliders in the upper right). The correlation is poor for red vectors, and excellent for blue ones. Vectors with anomalies detected in the civ processing (vec_F different of 1) are indicated in black (if the default option show black is checked at the upper right) . You can check the values of vec_F for each vector by selecting it with the mouse: press the pushbutton select with the zoom option on). When selected the vector color becomes magenta and its properties appear in the text window at the upper right (index in the file, x, y coordinates, u, v, and flag values in case of anomaly).

A common kind of anomaly is the systematic presence of black vectors in regions of large velocity, with values vec_F=-2. This indicates that the search box is too small, and CIV1 has to be run again with a larger search box (if the vector field looks good), or a smaller time interval dt (if many erratic vectors are observed).

After image quality and civ1 processing have been optimized, move to next step.

Note: The statistics on correlations vec_C provides in principle a convenient tool to optimize the time delay dt. For small dt, the correlations are only due to image noise (low luminosity) and possibly vibrations/fluctuations of the laser sheet. For larger dt, correlations decrease due to local turbulence and particle motion out of the laser sheet, as well as strain or rotation effects. The statistics on vec_C over a (rectangular) domain of interest, and time interval, can be obtained by the option stat_vs_time.

4.3 Remove false vectors(fix):

Use the fix_vel option in uvmat to remove vectors with low correlation (red vectors), and black vectors. Before running fix_vel, adjust the correlation threshold in view_field, so that red vectors are mostly erratic. A threshold about 0.3 is generally appropriate. In fix_vel, you should keep the vectors vec_F=2 (Hart) which are not precise but provide a valuable preliminary estimate for CIV2.

To check the result, you should come back to view_field. The removed vectors appear in magenta, or they are not seen if the option see_false (upper right) is off.

Note that this 'removal' operation is reversible: it modifies a flag (FixFlag gets a 1 as the second digit, e.g. FixFlag=10 in the absence of other rejection criteria) but all the vectors are kept in the files. So you can run again fix_vel with different criteria (for instance different thresholds on correlations).

Hand removal is also possible (but not advised for a long sequence of fields!). Get the mouse cursor with the pushbutton select and left click to select vectors. Click the pushbutton record to record the changes. Such removal is then recorded in the netcdf file by setting to 1 the second digit of the flag vec_F. Therefore it is not erased by automatic fix programmes (which act on FixFlag). Inversely, if you manually select a flaged (removed) vector, and push record, you rehabilitate it (you set FixFlag=0).

4.4 Velocity interpolation (patch):

The patch operation is needed to get spatial derivatives and prepare CIV2. It provides an interpolation on a regular grid (interp) and an interpolation with smoothing (filter). Both fields are stored in the same netcdf file as the CIV1 results.

Run the patch programme by the web interface at address http://Z4/CIV/interpolation.htm. Select the .civ file and click processing options, which yields a new web page. Introduce the image organisation options like for CIV1 and the number of grid points on which interpolation has to be done in each direction: the programme puts grid points at the minimum and maximum x coordinates of the initial civ grid and divides the length in regular intervals. Choose by default the same number of grid points as in CIV1. Select the same image pair as for CIV1.

Then set the spline parameters. Because of computation cost, the spline is restricted to sub-domains (with smooth matching). The default number 400 of points for a sub-domain is a good compromise. The smoothing increases for increasing parameters ρ. Use the default value 1 as a first guess.

The result of patch can be checked by uvmat, selecting the fields interp or filter. You can see the difference between these two fields, and get the corresponding histogram, by selecting both options simultaneously. For an optimum smoothing, the difference between filter and interp should be of the order of the measurement error, typically 0.2 pixel. You can even use higher values, as this smoothing is later used only to estimate the deformation of the fluid elements, while the velocity vectors are determined again from the images at each civ iteration.

By looking at the difference between civ1 and filter, you can further eliminate false vectors. Run fix_vel, selecting both civ_1 and filter, to remove vectors such that this difference is beyond a threshold. Choose a threshold value about 1 pixel, depending on the field quality. The vectors removed by this criterium have a FixFlag with 1 as the second digit (e.g. FixFlag=100 in the absence of other rejection criteria). You can then run again the patch programme with this improved field, and possibly with a better adjusted ρ.

4.5 CIV2:

Open again the http://Z4/CIV , choosing the option CIV2. Keep the same image pair (dt) as in CIV1. Activate the decimal shift and deformation options. The other options are not implemented, and Hart has no effect. The search zone is set by the programme to 2 pixels around the previous estimate (so the Isx,Isy and shift parameters have no effect).

Then you can check the results with uvmat, option CIV2. Bad vectors are now indicated by the flag vec_F=4, stating that the programme did not find a correlation maximum inside the search region. You can remove these bad vectors by fix_vel, and then use patch as for CIV1.

4.6 Further iterations:

You can repeat the same operations in an iterative process until convergence to a velocity field with no false vectors. If a filter2 field exists, which means that the previous CIV2 has been validated, the third CIV result is then recorded as "CIV2", while the former CIV1 result is erased and replaced by the former CIV2 result (if a filter2 field does not exist, it means that the previous CIV2 is not correct, then the new CIV2 result replaces the previous one).

So only the last two iterations are conserved with the respective labels CIV1 and CIV2.

If convergence to a velocity field with no false vectors is obtained, it means that a local correlation maximum at each measurement point has been reached, and these maxima smoothly vary with position: the difference with the filtered spline interpolation should have a quasi gaussian probability distribution with r.m.s about 0.2 pixel with no “tail” of large deviations.

Once this is obtained, you can run a new CIV2 with a longer time interval dt. Thanks to the previous estimate from smaller time interval, and the use of local deformation effects in the CIV2 algorithm, you can push the time interval further than with CIV1 (provided most particles do not leave the field by the transverse velocity). Then the precision will be improved. Furthermore the comparison of the results with the two time intervals, with independent image pairs, provides a convenient estimate of the measurement errors. The new result is then stored as CIV2 in a new netcdf file, for instance ..._ac.nc instead of .._ab.nc (and the previous result CIV2 in _ab.nc is then copied as CIV1 in _ac.nc. You can compare results from two different files in uvmat using the pushbutton “-” on the upper left.

If a high spatial resolution is needed, you can run a new CIV2 with a reduced pattern-box size, and an increased resolution of the measurement grid. The available previous estimate indeed allows to minimize the occurence of false vectors. Using uvmat, you can compare this new CIV2 result with the previous one, now labelled CIV1. A new CIV2 iteration would then erase the results with the previous pattern-box, so it is advised to save your previous results in a different directory.

As a summary, here is a typical sequence of processing:
file 10avril/dd/dd.civ, fields 1-118.
grid mesh for CIV:10 pix in x and y, 100x100 for patch.
Shiftx=Shifty=0 for Civ1, $\rho_{corr} = 1$
region of interest for statistics (statseries): x12=[120 220],y12=[20 200]

process	Ibx	Iby	Isx	Isy	ρ_{patch}	C_{min}	$diff_{max}$	rem vec	test file	rmq
CIV1	21	21	37	37	—	—	—	—	—	Isy too short (vec_F=-2)
CIV1	21	21	37	41	—	—	—	—	—	vec_F=-2 only for false vec
fix_vel(CIV1)	—	—	—	—	—	0.25	—	380	statseries1 (CIV1)	CIV1 OK
patch	—	—	—	—	1	—	—	—	—	filter-interp too small (rms 0.05)
patch	—	—	—	—	5	—	—	—	—	filter1 OK
fix_vel(filter1-CIV1)	—	—	—	—	—	—	2	160	statseries2 (filter1-CIV1)	
patch	—	—	—	—	5	—	—	—	—	
CIV2	21	21	—	—	—	—	—	—		
fix_vel(CIV2)	—	—	—	—	—	0.25	—	150	statseries3 (CIV2)	
patch	—	—	—	—	5	—	—	—	—	
fix_vel(filter2-CIV2)	—	—	—	—	—	—	1.5	55	statseries4(filter2-CIV2)	
patch	—	—	—	—	5	—	—	—	—	
CIV2 (3rd iter)	21	21	—	—	—	—	—	—	—	
fix_vel(CIV2)	—	—	—	—	—	0.25	—	25	statseries5(CIV3)	CIV3 OK

The progress of the results is monitored by noting the mean number of vectors removed at each fix_vel operation (in the text display window of uvmat), as well as the statistics (statseries file) obtained by running the option stat_vs_time of uvmat. The histogram of the image correlations (Chist) is particularly indicative, as well as the histogram of the velocity differences filter-civ.

5 Analysis of uncertainty and resolution

5.1 Particle drift:

Consider a simple model of a particle undergoing Stokes friction in an unsteady flow u(t).

$$\frac{du_p}{dt} = -(u - u_p)/t_F$$

, where $t_F = (2/9)a^2/\nu$=0.008 s, is the ratio of the Stokes force $6\pi\eta\nu u$ to the particle mass. If u(t) is assumed a pure oscillation $u = u_0 cos(\omega t)$, we find a relative error $(u - u_p)/u = (i + 1/(\omega t_F))^{-1}$. For a wave at frequency 1 Hz,

$\omega = 2\pi$, it yields a relative error 5 %. Since we are studying lower frequencies, we can consider that the particles follow well the fluid motion.

Particle sedimentation by gravity could be a direct source of error, but it is mostly an indirect one, by decreasing the density of particles in the field of view. The speed of sedimentation is given by the balance between weigth minus Archimede's force and Stokes force ($6\pi\nu\rho a v_s$), where a is the radius of the particle. We obtain then the speed of sedimentation $v_s = (2/9)ga^2(\rho_p - \rho)(\rho\nu)^{-1}$. For a relative density difference $\rho_p - \rho = 10^{-4}$ and a radius $a = 150\mu m$, the speed is equal to $5\mu m/s$ or 5cm in 3 hours.

5.2 Laser sheet position and width:

To be completed... Experiments with laser sheet using the diodes.... Light absorbsion without particles, with pliolite particles with known density, with polystyrene particles with known density....

Transverse profile of the laser sheet at different distances without and with particles (formation de halos par diffusion).

Light refraction by stratification:

5.3 Analysis of peak-locking errors:

Since the intercorrelation function is determined on the discrete pixels, the determination of its maximum is biased toward integer values. This shows up in the global probability density function of the displacements, which displays local maxima at integer values, see example in fig. 11.

We can estimate the peak locking error from such a pdf.

6 Matlab programmes for image reading and post-processing:

Reading a raw image:

```
%Reading of an image in 16 bits with nx*ny pixels:
    function [U]=readimage(nx,ny,path);
    fid=fopen(path,'r');
    B=fread(fid,Inf,'int16',0,'ieee-le');
    U=reshape(B,Nx,ny);
    fclose(fid);
    figure(1); %display
    imagesc(U');
    end;
```

Reading a velocity field (.u or .v)

```
clear all;
```

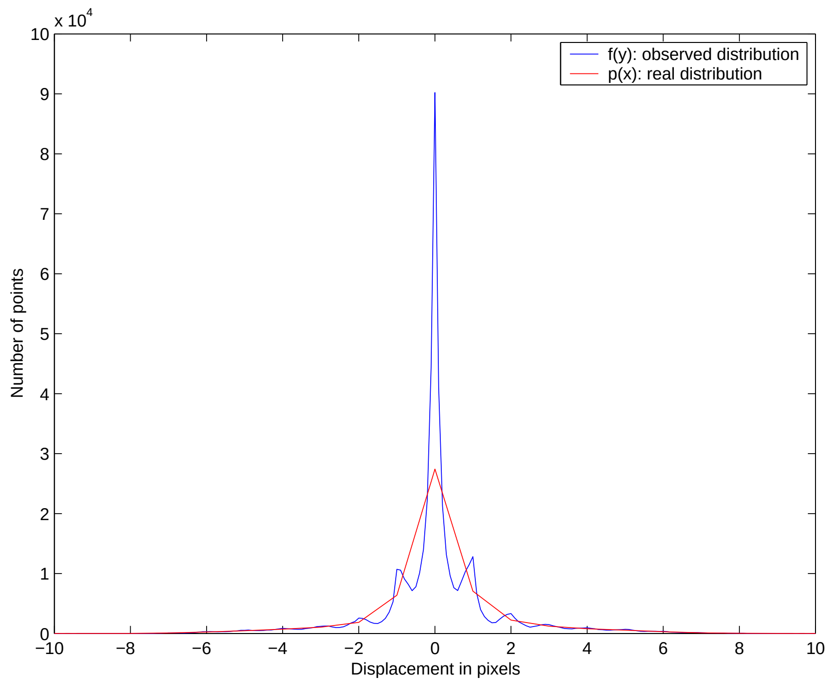

Figure 11: Probability density function of displacements obtained by CIV. To get a good statistics, we have sorted in bins of 0.1 pixel, the 50x50 x-displacements from a set of ... images (corresponding to the instability of a vortex).

```
%Lecture of a binary file .u
nx=50; %number of grid points in x
ny=50; %number of grid points in y
image=input('Please give the image name ending by .u :','s');
fid=fopen(image, 'r');
A=fread(fid, 1, 'int');
B=fread(fid,'float');
Nb_points_tot=A;
NB=length(B);
xx=B(1:4:NB); % x-coordinate
yy=B(2:4:NB); % y-coordinate
uB=B(3:4:NB); % radial velocity
vB=B(4:4:NB); % tangential velocity
fclose(fid);
U(1:nx,1:ny)=0;
a=1;b=ny;
for i=1:nx
U(i,1:ny)=uB(a:b)'; %build a matrice nx*ny
a=a+nx; b=b+ny;
end;
figure(1)
contourf(U,10); % example of representation of the radial velocity
```

displacement histogram:

6.1 Interpolation