STEREOMETRIC RECONSTRUCTION OF THE FREE SURFACE ASSOCIATED TO A FLOW OVER THREE-DIMENSIONAL ANTIDUNES

A Thesis
Presented to The Academic Faculty

by

Damien Douxchamps

In Partial Fulfillment
of the Requirements for the Degree
Electrical Engineer

Faculty of Applied Science
Catholic University of Louvain
Louvain-la-Neuve, Belgium
June 1998

STEREOMETRIC RECONSTRUCTION OF THE FREE SURFACE ASSOCIATED TO A FLOW OVER THREE-DIMENSIONAL ANTIDUNES

Advisors:

Pr. Benoît Macq

Pr. Piotr Sobieski

Ass. Christophe Craeye

Ass. Hervé Capart
(Civil Engineering Department)

Then Ulmo answered:
'Truly, Water is become now fairer
than my heart imagined,
neither had my secret thought
conceived the snowflake,
nor in all my music was contained
the falling of the rain.
I will seek Manwë,
that he and I may make melodies
forever to thy delight!'[...]
Thence he governs the flowing of all waters,
and the ebbing,
the courses of all rivers
and the replenishment of springs,
the distilling of all dews and rain
in every land beneath the sky.

J.R.R. Tolkien in 'The Silmarillion'

Acknowledgements

I would like to express my sincere gratitude to Hervé Capart and Christophe Craeye for guiding me through this research with exceptional patience and willingness to answer my numerous questions.

I am also grateful to Mourad Bellal and Olivier Cantineau for their precious help with the hydraulic system and the cameras.

Writing this manuscript with LaTeX would not have been possible without the help of Daniel Firre and Christophe De Vleeschouwer.

I address special thanks to Dimitri Devriendt who sucessfully conducted the velocimetric analysis of the flow.

Contents

List of Tables

List of Figures

Summary

The present report describes digital imaging techniques aimed at the experimental characterization of the free surface of a flow propagating over three-dimensional antidunes. Direct methods exist for the recovery of water surfaces, like resistive gauges. However, the unstable aspect of the flow we analyze is a source of great sensitivity to intruding objects. Our study will therefore focus on a non-invasive technique based on stereometry to recover the fluid surface from digital images of the flow. Another independent analysis based on velocimetric measurements was conducted simultaneously. It is subject to a parallel discussion in [7]. The collaboration of the Civil Engineering Department and the Telecommunications and Remote Sensing Department will make a cross-validation of both approaches possible, and hence provide a certain insurance in the quality of our results.

Part I

Introduction

Chapter 1

The fluid mechanics problem

1.1 Antidunes: from Nature to theory

The fluid mechanics phenomena we will analyze, though very common, is not trivial. A simple question will immediately introduce the reader to its complexity: why aren't river beds and deserts flat? One with no prior experience in the field might logically conclude that a turbulent flow over an erodible bed composed of granular material would obliterate any feature of the bed surface, and that the flat bed is the only stable configuration. Through regions of the bed should be progressively filled (due to the lower fluid velocity, stresses and hence sediment transport rate) with sediments conveyed from the crests, where the transport agents are more intense.

It is commonly observed, however, that sand beds deform under the action of turbulent flows into a persistent array of patterns, globally called bedforms. These can be created both by waves or currents. Wave bedforms can be observed on the beach, where the ebbing of shattered waves on the sand produces well known patterns (see picture A.1 in the appendices). Other bedforms, created by currents, have been mostly analyzed by [2],[3] and [4]. The short discussion presented here is strongly inspired from these authors, whom we recommend for additional informations. Current bedforms show very different patterns along the range of flow regimes. Among these, the three-dimensional antidunes we analyze occur at higher velocities. The shape of these antidunes also vary greatly within the considered regime of flow, from long rolls to peaky crests, with a range of complex patterns between those two extremes.

Though the theoretic problem is in itself very interesting to analyze, we can still aim its study to practical applications. These concern the determination of the effective water height during sudden rises in the water level of rivers, or the characterization of the fast flows that occur in dam-breaking.

1.2 Regimes classification

As yet, there is no complete mathematical model for the bedforms and the phenomena is not fully understood. However, scientists have been able to classify the different patterns, as functions of the fluid used, the sediment type (mainly the grain size), the geometric properties of the flow (slope,...), the flow rate, etc...

Kennedy [2] and Reynolds [3] have developed interesting models for the stability of the different bedforms. We will not describe their full theory, but focus on their results and on their consequences in our work. Following Kennedy and Reynolds, we use two numbers to characterize the flows. We first point the Froude number F_r, which is defined as

$$F_r = \frac{\tilde{U}}{\sqrt{gh}} \tag{1.1}$$

where $\tilde{U}$ is the average fluid velocity, g is the gravity and h is the average depth of the flow. Kennedy also introduces the quantity δ. At least two factors contribute to its value: the phase shifts between the bed displacement and the longitudinal distribution of the local flow properties, and the transport relaxation distance [1]. Kennedy showed the relation between the δ quantity and the wavelength of the patterns, and introduced a normalized value of δ:

$$j = \frac{\delta}{h} \tag{1.2}$$

The different regimes are shown on fig. 1.1 as functions of F_r and j.

1.3 Three-dimensional antidunes

Since our major concern is the three-dimensional antidunes, we will briefly describe the work of Reynolds. Kennedy only used the longitudinal wavelength of the bedforms to describe them, hence the three-dimensional antidunes zone on fig. 1.1 is biased. Three-dimensional bedforms must also take into account the transverse wavelength, λ_z, measured across the stream. Since the notion of wavelength is relative to the sediment grain size, Reynolds also introduces a normalized factor λ_z/D where D is the sediment grain diameter. Fig. 1.2 shows the different bedforms stabilities for two values of λ_z/D, where $M = \tilde{U}/\sqrt{\tau_0/\rho}$, τ_0 is the bed shear stress and ρ the water density.

[1] In fact, as stated by Kennedy: 'Perhaps δ can be placed in most meaningful perspective by considering it to be the factor of ignorance, in to which we lump those ingredients that are essential to the analysis but which cannot presently be calculated'.

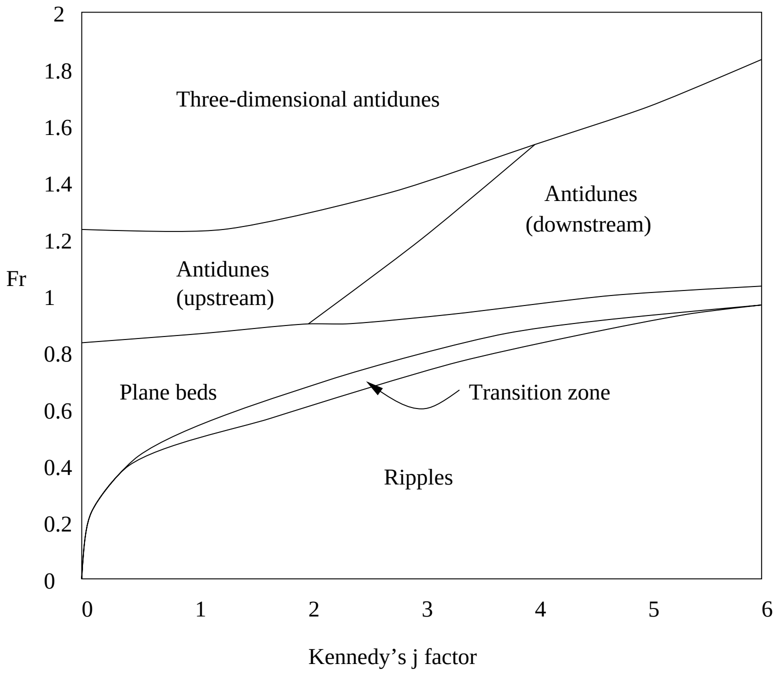


Figure 1.1: The stability zones for the different regimes, according to Kennedy.

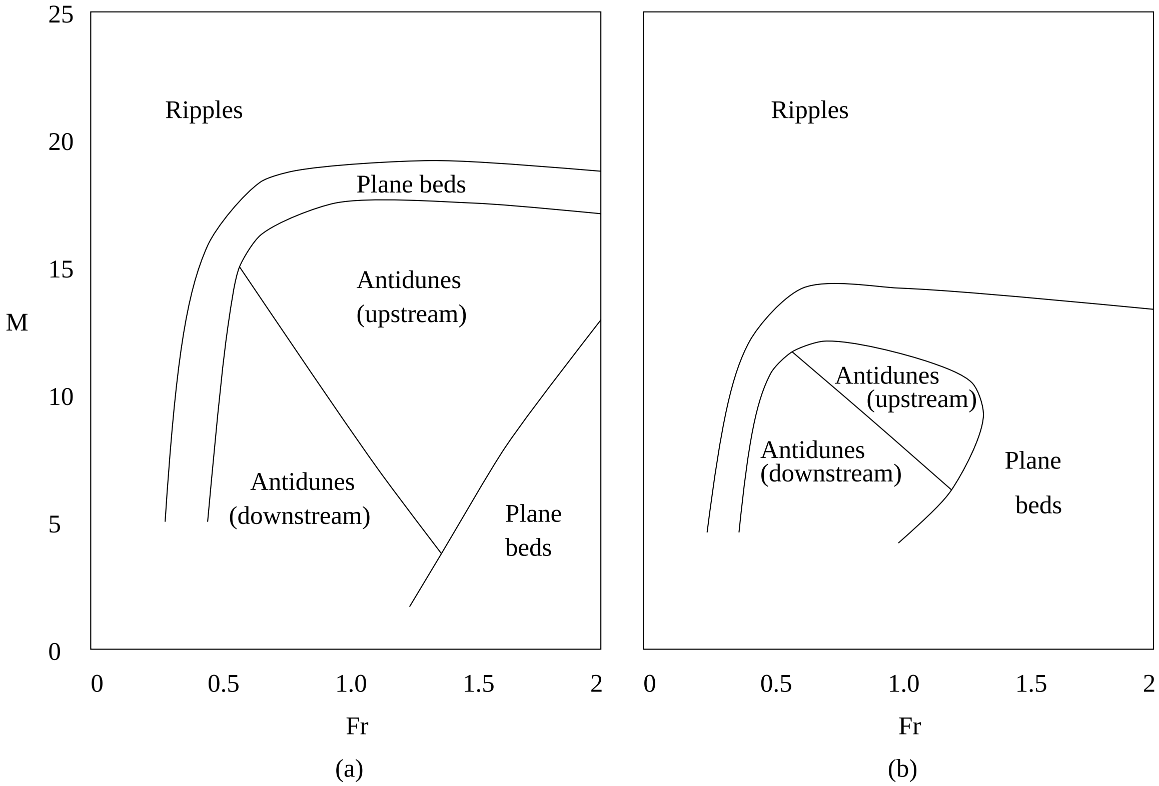


Figure 1.2: The stability zones for the different three-dimensional regimes, according to Engelund & al.:(a) λ_z/D =2000 and (b) λ_z/D =100.

One can see that in the three-dimensional case, ripples exist over a much wider Froude number range as λ_z/D lowers. This effect, together with the encircling plane beds zone, leaves a very small space for antidunes. In our case, the grain diameter was around 2mm, with transverse wavelengths the order of 20cm. This yields a λ_z/D of 100. The Froude number F_r we used was around 1.2 ($\overline{U} \approx$ 1m/s, $h \approx$ 5cm), which places us in the antidunes zone of the fig. 1.2b.

1.4 Conclusions

The goal of this brief presentation is to show the complexity of the phenomena and the variety of associated bedforms. The problem faced by scientists to fully understand the appearance these bedforms is partly due to the difficulty of acquiring the three-dimensional surfaces (or bedforms, for relations exists to pass from one notion to another). Indeed, having the precise form of the surface might help to characterize and later model the flow.

The absence of mathematical models also makes it difficult for the scientist to place its experiments in the good operational conditions and interpret the surfaces, if ever available. This vicious circle was broken using two independent techniques to capture the surface: a velocimetric and a stereoscopic method. Because of their independence, we will use the comparison of their respective results as a proof of the quality of these two approaches.

Chapter 2

Overview of surface recovery methods

2.1 Fluid mechanics solutions

The problem of surface recovery is not necessarily related to image processing or stereovision. Several methods were designed to recover a surface from other measurements. We will first discuss punctual methods which yield measurements at discrete locations, the check the velocimetric method which is the base of the other part of this analysis, presented by Dimitri Devriendt in [7].

2.1.1 Resistive gauges

This is an invasive method: we insert resistive gauges by pairs in the water flow (fig. 2.1). The resistivity of a pair will depend on the water level, which creates a short at the level of the water surface. Since our flow is unstable and thus very sensitive to intruding objects, we cannot use this method here. In the case of our pearls, the inserted objects were moving with the fluid, but here the resistive plungers are static with respect to the fluid, creating much more surface distorsion.

2.1.2 Pressure gauges

In this case, we place pressure gauges on the flume bottom, and read the pressure variation with the height of the water column above it. The problem for our experiments was to drill a complete array of holes for the gauges in our flume. Moreover, this method cannot handle with fast flows because the time-response of the gauges is slow, due to their physical properties and the sediment layer above them. It is usually specified for large scale, slow evolving phenomena.

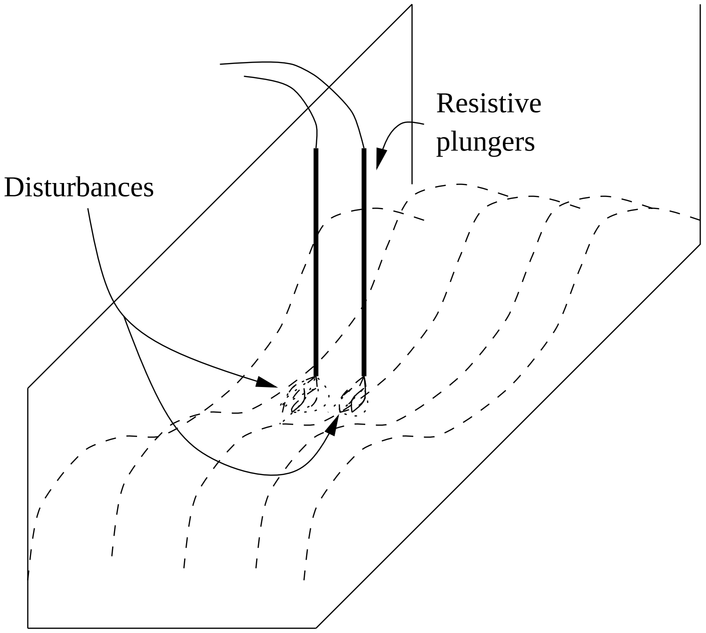


Figure 2.1: The resistive plungers approach.

2.1.3 Sonar

This non-invasive method consists in placing an array of sonars above the surface and compute the travel time of acoustic waves to the surface and back to the receiver. This system cannot be used in dense arrays because of interferences between devices, scatter, reflections…Given the complex surfaces present in our experiments, we need such a dense array, which prevented the use of this technique.

2.1.4 Particle based methods

An extensive study of those methods can be found in [5]. We will just introduce here the fact that both stereoscopic and velocimetric methods presented here and in [7] used particles as the elementary information available. Those particles consisted in wooden pearls and were tracked (in the velocimetric method) or matched (in the stereoscopic analysis).

2.1.5 Velocimetry

Extensive study of this problem can be found in [7]. We will remain here to a basic level useful to introduce and compare the different results. The indirect method proposed here relies on a simple measurement method which to our knowledge has not been exploited for this type of application. The development will be sketched here in a heuristic way, but can be demonstrated more rigorously.

2.1.5.1 Hypothesis

Basically, the behavior of streamlines particles moving slower over the crests and faster in the throughs is analogous to the motion of spheres rolling on a solid wavy surface: their movement will slow at peaks location and get faster in pits, due to the mechanic energy conservation:

$$E_{cin} + E_{pot} = constant. \tag{2.1}$$

It is only valid if the fluid viscosity is neglectable for in the other cases pearls would suffer friction at the fluid surface, thereby violating the energy conservation. Another hypothesis is that the flow is permanent, which implies that the trajectories cannot intersect with each other. We sometimes had intersecting trajectories for the particles, but these were few, and probably due to the non-zero size of the particles, while streamlines are immaterial.

From the point of view of pearls, we also assume that they follow the streamlines and that their speed is equal to the water speed. Due to the non-zero viscosity of the water, this is not always the case. We note however that these hypothesis are globally respected (see [7]).

There is a common hypothesis with the stereometric method illustrated below. It concerns the quasi-steady motion of the bedform shapes and hence of the water surface. This allows us to concatenate the data collected for several images, as far as the total acquisition time of the data is kept small with respect to the time necessary to see the bedforms change. In our experiments, the time needed for a complete run was below 3 seconds, which is very low compared to the speed of the bedforms.

Finally, we make the hypothesis that the bedforms are periodic in both the X and Y directions of the mean surface plane. This is a conclusion of the theory of Kennedy in [2] and has also been very well verified by our experiments.

The high number of hypothesis required for this method restricts its applications, but it yields an elegant and efficient way to recover the surface from a unique camera. Note that from these hypothesis only one, the quasi-steady surface, will be used for the stereometric approach.

2.1.5.2 Velocimetric measurements for height estimation

Since the bed morphology evolves only slowly with respect to the flow velocity, we can reasonably consider that the flow behaves in a quasi-steady fashion. Along a surface streamline (identical to the paths of particles since the flow is considered quasi-steady), neglecting dissipation an mean slope terms, we can express the Bernoulli equation as

$$\frac{\vec{V} \cdot \vec{V}}{2g} + Z = \text{constant along a streamline} \tag{2.2}$$

where $\vec{V} = (U, V, W)$ is the velocity and Z the altitude of a point of the water surface. We may consider that the flow over antidunes can be described, in first approximation, as small fluctuations around a mean uniform flow in the X direction and write

$$\frac{\tilde{U}(U - \tilde{U})}{g} + (Z - \tilde{Z}) \approx 0 \tag{2.3}$$

where the tilde denotes an average over the entire flow region. In derivating (2.3), we have considered that the velocity fluctuations following three directions are all of the same order and much smaller then $\tilde{U}$. Also, it is because a uniform flow along the X directions presents a horizontal surface in the transverse direction Y that we have been allowed to consider $Z \approx \tilde{Z}$. If the horizontal velocities are known along streamlines, expression (2.3) offers a way of estimating elevation with respect to the mean surface plane.

2.2 Image processing solutions

2.2.1 Pattern projection

This method consists in projecting a pattern of light on the object. The projecting device may be seen as an inverted camera. From this point of view, the problem is very similar to stereovision: if we know the parameters of the projecting device through calibration and the shape of the projected pattern (which is an a priori knowledge), reconstruction is possible from a single camera. We can use a dense two-dimensional pattern (e.g. two-dimensional pseudo random with good cross-correlation properties), or a more discrete pattern, with different recognizable features. More about this approach can be found in [18].

This method is very cost effective for opaque objects, because only one camera is required, and the projection system is usually cheaper than a camera. Also, since the pattern is projected continuously, the camera is the only discrete-time system and no synchronization problems between devices occur (see Chapter 9). If the object is static the projected pattern might be changing; a typical case is the projection of a moving ray of light on the object, described in [16].

We could not use this setup here because the water surface is transparent and reflective. The pattern seen by the camera would be the one on the bottom, distorted by its path through the water. Moreover, many parasitic reflections would occur. It is possible to use this system if the surfaces are very flat, so that the projective device and the camera can be placed at angles with the surface less than the water Brewster angle on the whole surface, and without occlusions.

2.2.2 Moving camera

For static objects, the cost of a second camera can be avoided by moving a single camera around the object. This gives numerous different views of the object, and thus good precision can be achieved through triangulation. Since views vary continuously, they differ very slightly from one to the next, and it is also easy to match and to track feature points.

In the case of a fast water flow, it is not recommendable for obvious reasons: to consider the water static with respect to the camera, the latter should be moving far too fast (100 m/s for 1% tolerance, since the fluid speed is $\approx$ 1m/s). One could argue that we do not analyze the water particles (moving fast) but its surface (quasi-steady). The problem is then to fix steady features to the fluid surface, which cannot be done without interacting with the surface and disturbing the flow.

Note that for static non-reflective objects, one can combine the projected pattern and moving camera techniques. The projected pattern is then static, and we move the camera to have more views of the object an enhance triangulation. In general, moving the camera around a static object will provide greater accuracy for triangulation. Some other work in the field include [25].

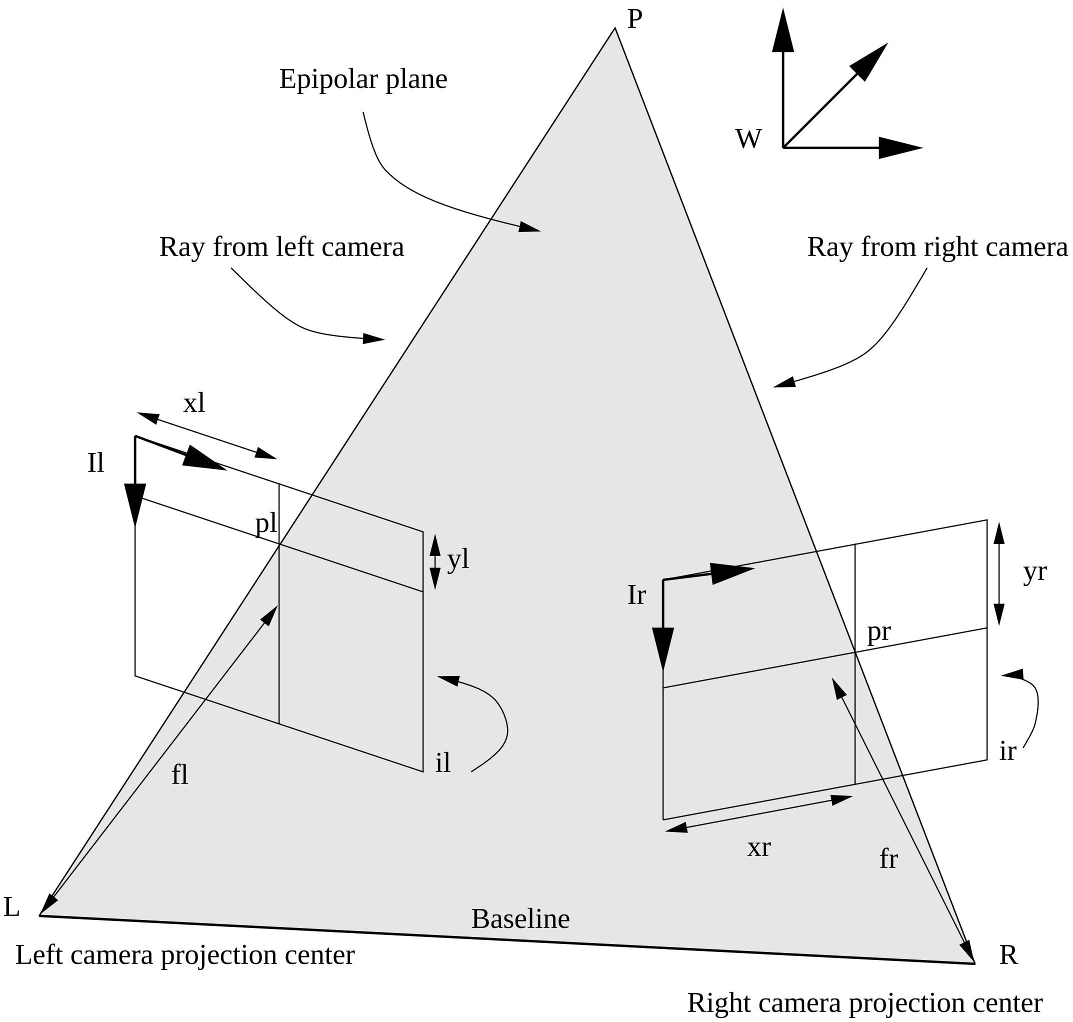


Figure 2.2: A geometric representation of the stereovision problem.

2.3 Stereoscopic analysis

2.3.1 Introduction

Stereometry consists in determining the depth of objects or some features of objects by using two cameras as the source of information. The geometry of the problem is shown on fig. 2.2. Each camera can be considered as a system which defines a central projection of the world onto an image plane. The two focal points of the cameras which are also the centers of the projections are located on L and R. The LR line joining the two centers is called the baseline. The image planes i_l and i_r correspond to the location of the image in the referential attached to each camera. Their distance to their respective focal center is called the focal length (f_l and f_r).

During the triangulation process, we will use the *rays*, defined as the lines passing by the point P and the projection center of a camera. Rays represent, in a way, the path of light from the object to the camera. Their intersection with the image planes gives the two images of the point P: p_l and p_r.

Five coordinate systems appears on the geometry:

- the world coordinate system, represented by its referential W. It is the coordinate system in which we represent three-dimensional objects, with coordinates expressed in millimeters.

- the two specific coordinate systems I_r and I_l for each image plane. The unit use for these referential is the pixel.

- the two three-dimensional coordinate systems associated to each camera (not shown on fig. 2.2). These are defined such that the image plane is located at a constant Z coordinate of 1, while the X and Y axes are the same as the image plane referential.

The point P being part of the 3D scene will have p_w as world coordinates and $p_l = (x_l, y_l)$ and $p_r = (x_r, y_r)$ as image coordinates.

The goal of stereometry is to provide the world coordinates of a point from the two initial image coordinates. It is not possible to do it with only one image plane, because it would be equivalent to determine three unknowns (the world coordinates) from two known values (the (x, y) image coordinates). However, with two image planes, we will have four known values (the two coordinates couples (x_r, y_r) and (x_l, y_l)), and the problem is overdetermined. The solution will thus be an approximation, usually the least mean square solution of the problem.

2.3.2 Process overview

The solving of this problem can be divided in several steps. We will briefly describe those here, but since this is the technique used throughout this work, more can be found in later sections.

First, we need to know the parameters (position, angles and focal length) of the cameras in the world coordinate system. This is the calibration process, by which the geometry of the problem is determined. It requires to have a special set of three-dimensional points for which the two image coordinates *and* the world coordinates are known (see Chapter 5).

Once the calibration is achieved, we want to locate feature points in the images. The process must be able to locate 'interresting' points, like corners, edges,... It can be done through classic algorithms like wavelet transform [10], mathematical morphology [22], edge or feature points detection [1],[24].

Then follows the correspondence step. We supposed that we had the image coordinates p_l and p_r for the point P, but while it is relatively simple to locate these separately, being able *match* these feature points (i.e. say that one point on an image plane represent the same three-dimensional point as another point on the other image plane) is more difficult.

This step requires that we have special properties for feature points, independently from the direction from which they are observed. Such properties could be texture, neighbourhood, edges, corners,... With each point bearing some caracteristics, it is possible to match them. An example of what can be done with this method can be found in [19], while a general approach of the problem is described in [20]. We may also point the original Moiré method presented in [15].

Another approach, which we followed, is to base the matching not on point properties, but on the geometric parameters. This is necessary in our case because all the pearls look the same and have no particular distinct features.

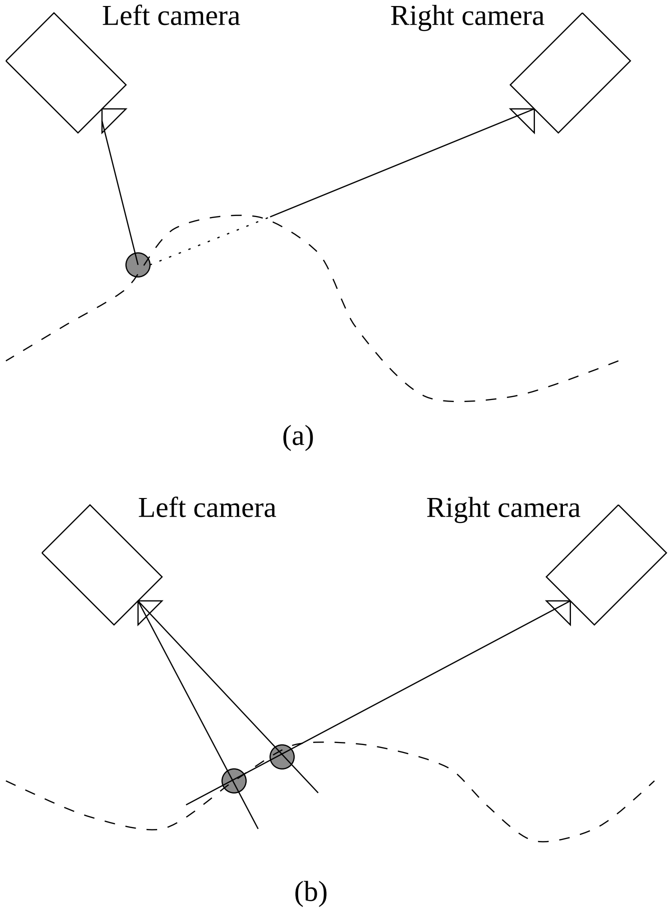
2.3.3 The occlusion problem

We here illustrate the difficulty to match the left and right views by briefly introducing the reader to the occlusion problem, which concerns the visibility of certain points on one image plane and not the other. Because only one image coordinate is available for that point, no reconstruction of its three-dimensional coordinates is possible. This is illustrated on fig. 2.3a, where the particle is hidden by a fluid crest.

A different kind of occlusion can also occur when two points, though having different image coordinates from one point of view, are superposed on the other view (i.e. they have the same ray, as shown on fig. 2.3b). Most of the time, both points are unusable.

A last problem is the fact that some feature points might simply be out of the field of view of one camera. Since these points do not have one representation on each image plane, they should also be rejected.

From these example the reader will understand that views are usually quite different, even for simple objects. In general, views will be similar if the angle between



(a)

(b)

Figure 2.3: Two kind of occlusions: (a) classic and (b) mutual.

cameras is low, and the baseline is small. Unfortunately, this will lead to a greater error on the depth, because we extract depth from the differences between views. If these are small, and also because the images are discrete, the error will become larger. On the contrary, greater divergence between the views will make feature points difficult to match, but if a match occurs, the precision will be higher. This compromise depends on applications. In our case, we have chosen several configurations for the cameras, looking for the best solution. These will be compared later on in Chapter 10.

We now briefly present the approach used to process the first experiments.

2.3.4 Brute force matching

The brute force matching technique tries to match portions of the first image plane with others in the other image plane. In short, we convolve one entire image plane with a portion of the other plane, and look for the maximum. This maximum is supposed to be located where the two parts best match, i.e. were they are the most likely to contain the same feature pattern. This might yield good results, but requires a huge computing time.

The most classic advancement is to recompose the images so that lines in both image planes are parallel to the baseline and match each other (i.e. line x on the left image plane correspond to line x on the right image plane). This process, called epipolarization or image rectification, transforms images into rectified images. It could be seen as the projection of the two images on a plane parallel to the baseline.

With this transformation, a line in the first image plane corresponds to the same line on the other image plane. If we work with these views, the search for the match in the second image plane can thus be restricted to a unidimensional search along a line of the rectified image, because the match between lines is already done. The complexity of the problem is $\mathcal{O}(n)$, while without epipolarization it was $\mathcal{O}(n^2)$.

This technique, presented by [21], was at first used to process our experiments because it was simple and we could easily see whether the disparities were significant and thus exploitable. Since the water surface is transparent, we had to attach a feature pattern to it for matching. We thought about using a mixed series of black and white pearls that would create a random pattern on the surface, which we hope would have good cross-correlation properties. It appeared however that results were rather bad, for the following reasons.

A problem with this approach is that having a pattern with good correlation properties requires a large matching block, and thus more computing time, because particles are themselves rather large (10 pixels). Moreover, if the block is larger, its properties become global (e.g. the size of a whole crest of the fluid surface), not local. Trying to determine the depth with portions of the image where the depth is far from constant would yield an unrevealing surface. This idea has thus been rejected as a possible complete analysis method.

Note that brute force matching was developed supposing a dense depth map, i.e. a depth map computed for each point of the image planes. One can in most case afford a sparse depth map, with depths available for selected points (e.g. on a grid). Though it reduces computing time in a significant way, all other problems remain.

Part II

Experiments

Chapter 3

General setup

After this quick introduction to the fluid mechanics problem and some of its solutions, we will focus on our experimental setup. This will provide the reader sufficient information to capture the global geometry of the problem. Since stereovision and geometry are closely related, this point is important and is abundantly illustrated in Appendix B.

During this discussion and the following developments, we will speak of two series of experiments. They were both made at the hydraulic facility of the Civil Engineering Department of the Catholic University of Louvain-la-Neuve, Belgium.

The first runs were conducted in November 1997. Their goal was to become familiar with the experiments hardware and software, provide a first series of data for primary analysis and understand were the major constraints and problems were, so we could solve these before the second series of runs.

No	Date	Flow rate [valve turns]	Surface type	Available images
0	Nov 1997	n/a	complex pattern	1
1	Feb 1998	3	complex pattern	none
2	Feb 1998	3	complex pattern	none
3	Feb 1998	2.5 to 3.5	unidimensional (rolls)	50
4	Feb 1998	3 down to 2	peaks	15
5	Feb 1998	5	complex pattern	46
6	Feb 1998	3	complex pattern	55
7	Feb 1998	2 to 4	complex pattern	none

Table 3.1: The different runs and their characteristics.

By fall January 1998, we had developed a complete set of software procedures to handle the processing of the data. The first reconstructed surfaces were obtained and published [6]. These first attempts comforted us in the possibility to recover the water surface both from stereoscopic and velocimetric processes. A second set of experiments was launched fall February 1998. These runs provided much of the data presented here. Table 3.1 describes the different runs, with their characteristics. The February runs also included simultaneous velocimetric measurements for methods comparisons and therefore except otherwise noted, we will only describe the latter runs. In further discussions, we will identify the runs by their number, most of the time run 3 to 6.

Several aspects of the flume facility will be presented. We first focus on the flume itself and its associated hardware and hydraulic system. In Chapter 4 we will talk about the cameras and the acquisition systems.

3.1 The flume and the hydraulic system

A general scheme of the system, reprinted from [8] is provided on fig. 3.1. Experiments were made in a square flume, 500mm×500mm in section and five meters long. The slope of the flume (1%) was chosen as to yield antidunes for the envisioned regimes. The flume is partially filled with an average height of five centimeters of sediments.

Water is introduced from an upstream reservoir in the flume. The reservoir is filled at constant rate, achieved by an overhead tank filled at a constant level, creating a fixed height water column. The level of the reservoir is determined by an overflow pipe. The water which passes through the flume then reaches a filtering system to prevent sediments to reach the circulation pump. Pearls injected on the surface for analysis are also collected with a cage at this stage. The filtered water is sent to a lower reservoir, where the circulation pump takes it back to the overhead tank, in a closed loop.

Since sediments are washed by the flow during the experiments, we have to import a fresh stock of these during the runs to avoid the sediment level in the flume to drop along the runs. This is achieved using a silo located in the upper part of the flume. The sediments there flow through evenly spaced holes on the whole width of the flume. One should note that fresh sediments must be perfectly dry to avoid a arch-effect in these holes, which would create irregular feeding and eventually complete jamming. It is thus necessary to dry used sediments in an oven before reintroducing them. This process takes hours, and was conducted during the night. Hence the number of experiments during a day is mainly limited by the quantity of fresh sediments

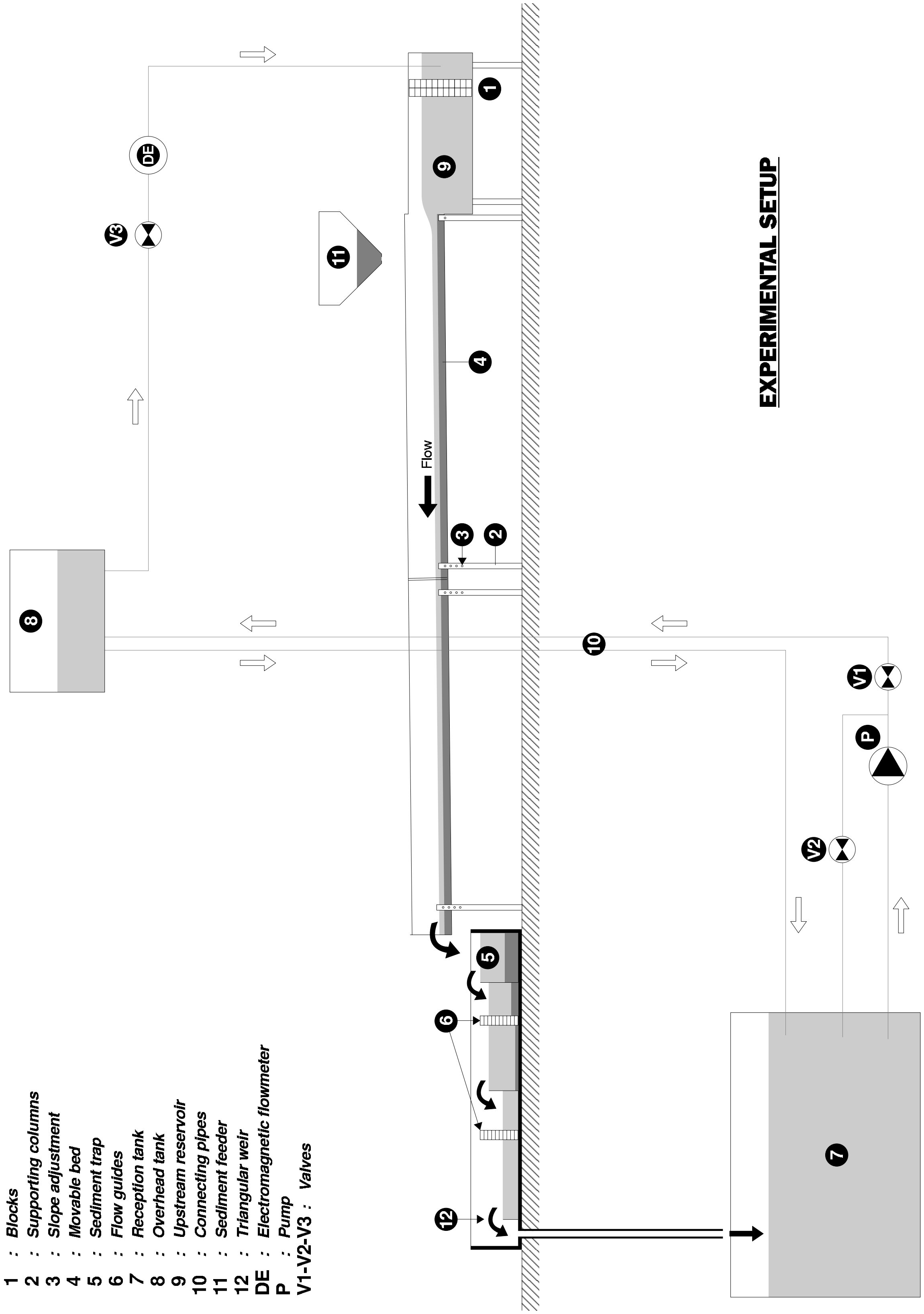


Figure 3.1: The hydraulic system.

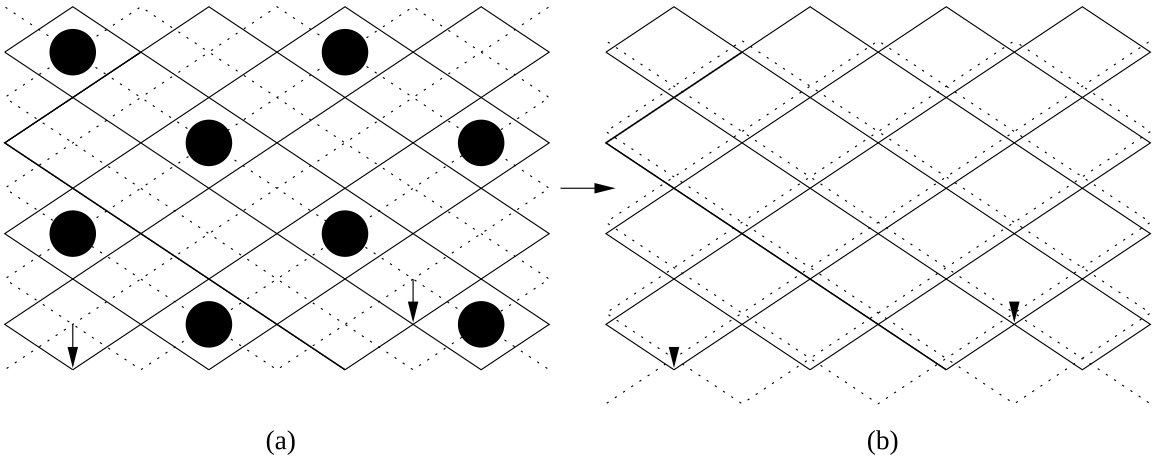


Figure 3.2: The spreading system using sliding grids: (a) when refitting, (b) when releasing.

available and the capacity of the drying oven.

3.2 Pearl spreading systems

Both the stereoscopic and velocimetric methods use floating tracers on the water surface. The use of tracers supposes an appropriate system for spreading them evenly across the duration of the experiment and the width of the flume. Two system were used, with different efficiencies.

The number of tracers used was 1000, which is a compromise between a higher number of points for surface reconstruction and a more difficult matching and detection. The tracers consisted in wooden pearls, 9 mm in diameter, painted in black or white depending on the experiments.

3.2.1 Sliding grids

This system used the superposition of two metallic grids. They were placed above the flume (some 40cm above the water surface). When slightly shifting the upper grid with respect to the lower one, its holes are obstructed by the lower grid and pearls can be spread evenly across the grid before the run (fig. 3.2a). When the antidunes are formed, re-aligning the two grids then let the pearls pass through (fig. 3.2b), all at the same time, creating a regular pattern on the water surface. However, due to the properties of the flow, pearls concentrate at peaks and tend to avoid valleys and the flume boundary regions. This artifact could not be avoided since it is based on

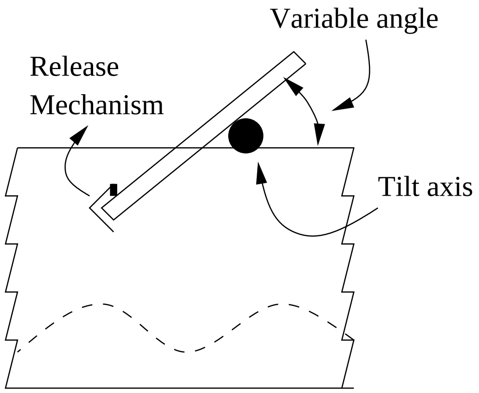


Figure 3.3: A section of the flume showing the panpipes-like spreading system.

the intrinsic flow properties (i.e. the energy conservation). This system is thus only used to provide a good general spreading at the start of the run, not to achieve a perfect pattern on the fluid surface.

The practical problem with this setup is that one must spread the particles evently on the metallic grid, which is quite time-consuming. Another system, which does not suffer this kind of drawback has also been tested.

3.2.2 Panpipes-like

In this alternative, a panpipes-like system was placed across the flume width. Pearls were placed in the tubes by their upper extremity and the lower outlets were bloked by a sliding plate (fig. 3.3). The panpipes could be tilted to an appropriate angle, so the flow rate of pearls (hence their spread across the surface) could be controlled. Once the antidunes are formed, we remove the plate and let the pearls fall freely on the surface. Replacing the pearls is much more easy but the system suffers another problem: while dry pearls will effectively fall correctly, wetted ones will stick on the inner side of the tubes. This is reinforced by the fact that the tilt angle of the system was very close to horizontal so that a wet pearl was able to bloc an entire tube. Increasing the tilt angle did not produce better results: pearls were still stucked in the tubes, and their fall was rather erratic so this system was abandonned.

Chapter 4

Acquisition techniques

Two different kinds of cameras were used, according to the type of measure envisioned; for each one, we will discuss the operational conditions. The two first sections describe the separate use of velocimetric and stereoscopic methods (related to the November runs). We then show how to make simultaneous use of both techniques and what compromises it implies.

4.1 Velocimetry

4.1.1 Cameras

For velocimetric measurements, we wish to measure the instantaneous velocity of particles which implies particle tracking. If the time lapse between takes is too high, the tracking becomes difficult, because the particle position can only be predicted over a small distance. This is due to collisions between particles, the crossing of trajectories,. . . One would then look for the highest frame frequency available. However, several constraints will limit the frame rate.

First, due to the limited frame memory available on the high-speed camera (800 frames), it was not possible to achieve the maximum frame rate of 1000 frames/sec during the whole experiment, which lasts 2 or 3 seconds.

From a velocimetric point of view, using a 1000 frames/sec is not particularly useful: the camera with a limited resolution cannot see a significant displacement of the particles during this time lapse. It would then be necessary to average the displacements over several frames.

The tracking problem can be easily solved if the movement of particles is less than their diameter during the lapse between frames (see [7]). Since the particles diameter is 9mm, and the flow speed is approximatively 1 m/s, we will need at least 1m/9mm=111 frames per second. The closer frame rate available on the high-speed

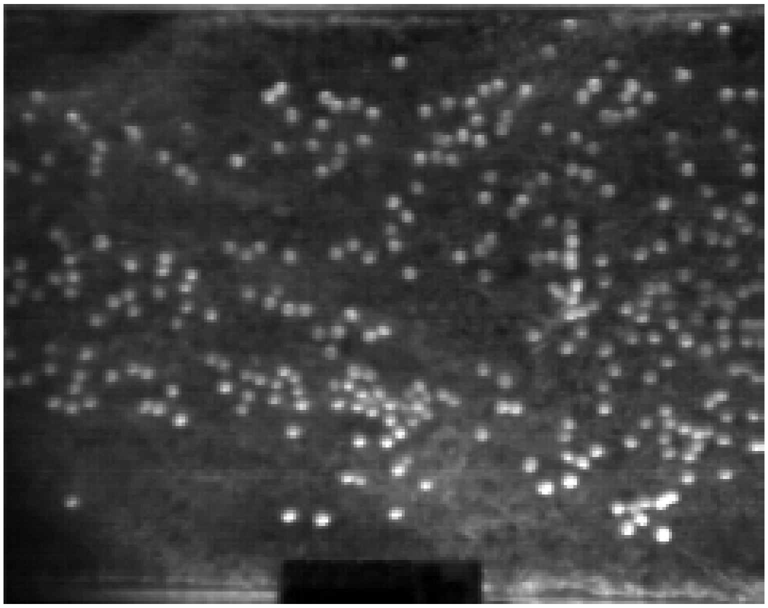


Figure 4.1: An image from the high-speed camera during run 5. The lower black rectangle is the counter used for synchronization.

camera is 125 frames/sec, but since the run lasts less than three seconds, we could afford a frame rate of 250 frames/sec. It was the best compromise with our Kodak Ektapro-1000 Imager.

The velocimetric technique only measures the speeds in the horizontal plane and it is thus important to limit as much as possible the influence of vertical speeds . Hence, the high-speed camera used for these experiments should be vertical. The lenses we had were such that we needed to place the camera at a very high altitude compared to the stereo setup: nearly four meters. The major problem with this setup is to build a stable platform at this altitude for the camera. Since our scaffolding was rather not of this kind, we took special attention not to disturb it during the runs and also wait for its complete stillness before starting a new run. The stability was improved by locking the scaffolding on the flume, which is a massive inert structure.

Using a fast camera could not be done without compromise. It has a small resolution (192×239 pixels) which limits its accuracy. Aside from this lack, advantages include easy file transfer (it was computer driven) and synchronizability. However, one should keep in mind that the price of such a camera is rather high (around 10.000 USD).

4.1.2 Lighting

The high-speed camera requires a considerable light power, since its sensitivity is low at the high frame rate envisioned. In the first experiments, we used two 2kW spots placed on each side of the flume. A tracing paper diffuser was placed on the flume boundaries, far enough from the light source to avoid burning. The light sources were placed below the water surface for minimal reflectance. This position also yields a darker sediment background and thus a better contrast with the white particles.

4.2 Stereometry

4.2.1 Cameras

For stereometric measurements, commercial digital cameras were used. Their only advantages compared to professional systems are low cost and quick availability, but we would not recommand them for most applications for the following reasons:

- they are almost never synchronizable, while this aspect is critical for stereometric measurements of a moving surface (see Chapter 9),
- users manuals are insufficient and lack precision for professional applications,
- most of the settings are qualitative, not quantified (shutter speeds, ...),
- some parameters have coupled effect when changing them,
- some settings also depend on the scene filmed when they are set. We found out that the range of shutter speeds available in manual mode changed with respect to the light intensity of the scene when the manual mode was set. Since we wanted a very low shutter speed, we needed to film a bright light source whenever setting the manual mode. This might be seen as a detail, but finding out how the camera reacts without proper documentation made us loose precious time.
- Other problems include auto power off (non disconnectable) and the lost of all settings when in sleep mode. Given that the sleep mode was automatic, it means that 'all settings are lost automatically'...

Half the runs we made were lost because of insufficient knowledge of the hardware. Still, they have the major advantage of their good resolution (576×768, a tenfold increase in pixel density compared to the high-speed camera).

For these early runs, the right camera was placed vertically and the left one adjusted for a maximal overlap of the fields of view. The distance between the cameras was small, about 30cm.

4.2.2 Lighting

The requirements are not so important compared to the high-speed system, since our frame rate is fixed to 25 frames/sec, ten times less than the Ektapro camera (note, by the way, that each camera takes the same amount of information each second). Two 200W spots were sufficient for a nearly uniform lighting. Given the clear sediment bed, black particle were used for optimum contrast. Hence the light sources were placed to illuminate the background rather than the particles and also below the water surface to avoid reflection.

The approach followed here is essentially experimental because theory in the field is very complex and difficult to apply. More about the theoretic aspect of lighting can be found in [17] and [23].

4.2.3 Synchronization

Even if we cannot synchronize the cameras, we can use the images provided that they bear a universal time tag. We therefore placed a counter in the field of view of both cameras. Each camera records its value on each frame as the absolute time when the image was taken. More about the synchronicity problems will be presented later.

4.3 Combining both methods

Though using each camera system separately is relatively easy, we will need to make compromises for the simultaneous use of the two methods. This aspect of our analysis is capital, since we want to see whether both methods yield the same surfaces, which implies measurements of the same run at the same instant, because the surface moves (even if slowly) and changes sometimes significantly from run to run given its unstable nature.

4.3.1 Setup

The general setup does not change, but the stereometric cameras cannot be placed in a vertical fashion like in the first experiments, because they would obstruct the field of view of the velocimetric camera. Since the high-speed camera must be vertical and cannot be moved, the stereo cameras were spaced to allow the velocimetric camera to see the same portion of the flume. Spacing the cameras also impose a greater tilt angle so that both stereo cameras see the same portion of the flume.

4.3.2 Lighting

When the two systems were used together the lighting, which was easy in the separate cases, becomes critical. Since the high speed camera was the most demanding, setups were first made for it, then commercial stereo cameras were adapted.

We then used four 2kW spots to have an optimal light repartition for the high-speed camera. Particles were also white according to the previous good velocimetric results of the November runs, leading to a worse contrast for the stereo system.

The major problems occurred with the stereo cameras. First, with the enormous amount of light in the flume, shutters were set to maximal speed. This caused the counter, placed slightly off-field, to be completely unreadable, because its location above the flume was not illuminated. The only solution was to dim the illuminated part of the scene to have comparable light levels between the counter LEDs and the flume. This was achieved using filters on the camera's assembly, but only shading the flow part of the scene. We could then read the counter and still have enough contrast in the flow area.

4.3.3 Synchronization

The fact that three cameras were viewing the scene at the same time was interesting for stereovision: we could use three cameras for triangulation instead of two. This improvement was intended to be exploited, so we looked for a synchronicity system between the three cameras. For the stereo system, it was already done by an external counter, but nothing had been envisioned as yet for synchronizing the velocimetric camera. Indeed, for the high-speed camera, we could not afford dimming a part of the scene to read the counter, which was also invisible. Since the time tag could not be present on the velocimetric images, we used the fact that the high-speed camera

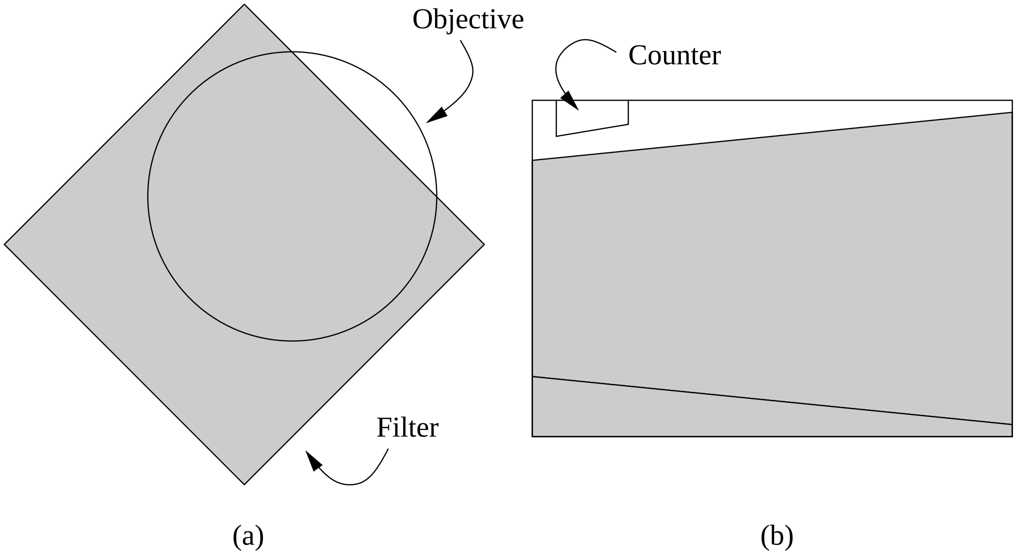


Figure 4.2: (a) The position of the partial filter on the stereo cameras assembly and (b) the resulting shading of the image.

can be synchronized.

The trigger of the external counter is therefore sent to the high-speed camera so that when the counter is started, it triggers the velocimetric camera. Each frame of the high-speed camera then bears a time tag, the first one is 0, the next .004 (because we have a frame rate of 250frames/sec), and so on. This time can be compared with the counter time since the delay between the two is zero.

Part III

The particle detection based stereometric process

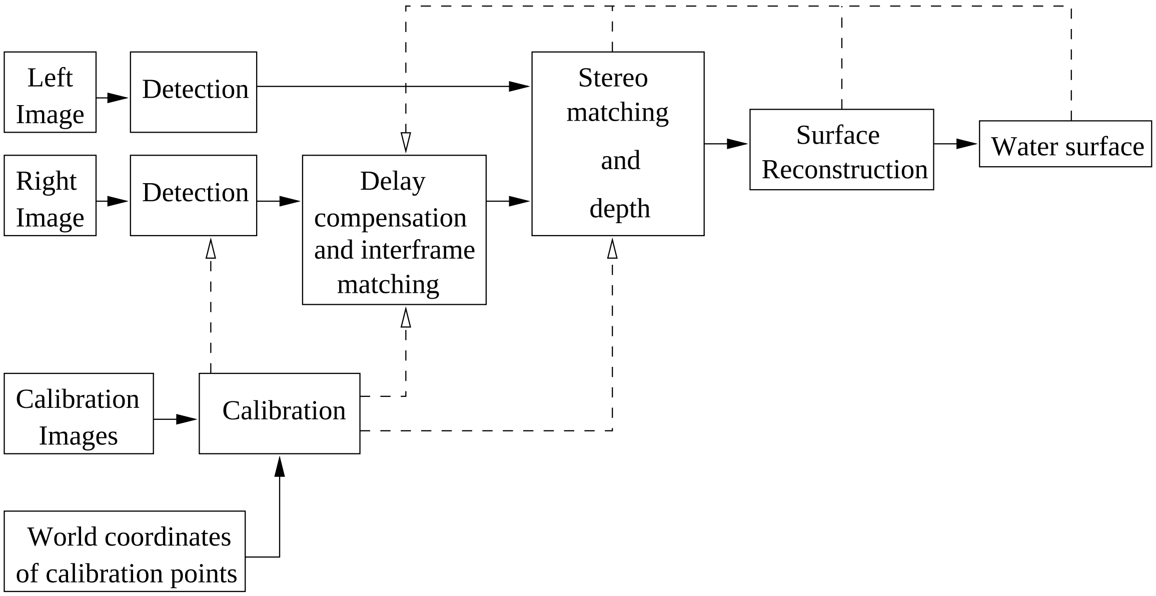

Left Image
Detection
Stereo matching and depth
Surface Reconstruction
Water surface
Right Image
Detection
Delay compensation and interframe matching
Calibration Images
Calibration
World coordinates of calibration points

Figure 4.3: A general view of the stereometric process. Dashed lines indicate parameters influence, while continuous arrows show the data path.

In this section, we present the stereoscopic method that has been used to process the image sequences. A general schematic diagram is provided on fig. 4.3.

The first step is to detect the particles. Once their location is estimated, we match particles in the left and right views, and compute the three-dimensional location of the point corresponding to each pair. At this point of the process, we have a list of 3D points supposed to belong to the surface. The latter is then reconstructed from these scattered points using regularization methods.

Before these steps, we will focus on the calibration process, by which we determine the internal and external orientations of the cameras. The fact that cameras are not synchronized will be discussed in the end.

Chapter 5

Calibration

5.1 Introduction

The stereovision system has been introduced as the determination of three-dimensional points from a pair of two-dimensional coordinates. The evaluation of the parameters of this essentially geometric operation is handled by the calibration.

We will present a first simple calibration method, then explain how it can be improved to take more internal camera parameters into account. These two aspects are mostly inspired by [1]. We will end this discussion by showing its practical implementation and how the necessary data for this step were collected.

5.2 A simple calibration method

This method is aimed at the evaluation of all the *external* parameters of the cameras (pitch, yaw and roll angles, plus its geographic position in the setup). Only one *internal* parameter (i.e. a parameter describing the camera itself), the focal length, will be determined. More internal parameters evaluation will be obtained through the affine calibration.

This simple calibration process will give the essential relationship between the image plane coordinates (x_i, y_i) and the world coordinates (x_w, y_w, z_w) of the scene points. Note that this is done independently for for each camera, hence we use the subscript c which can be replaced by l or r for the left and right cameras.

We now define a three-dimensional referential for each camera, so that a point (x_i, y_i) of the image plane has coordinates $(x_i, y_i, 1)$ in that referential. In camera coordinates, the equation of the ray passing through the point (x_i, y_i) in the image plane is thus $(x_c, y_c, z_c) = t(x_i, y_i, 1)$ with the c subscript indicating the three-dimensional camera referential and t going from zero (the center of projection of the camera) to infinity. With $t = 1$, the point (x_c, y_c, z_c) is the image plane point $(x_i, y_i, 1)$.

The way to obtain the equation of the rays in world coordinates is to apply a rigid body transformation. It consists in a rotation, expressed by the orthonormal rotation matrix R:

$$R = \begin{pmatrix} r_{xx} & r_{xy} & r_{xz} \\ r_{yx} & r_{yy} & r_{yz} \\ r_{zx} & r_{zy} & r_{zz} \end{pmatrix} \tag{5.1}$$

where the different terms can be expressed as functions of the angles ω (pitch), ϕ (yaw) and κ (roll):

$$\begin{aligned}
r_{xx} &= \cos\phi\cos\kappa \\
r_{xy} &= \sin\omega\sin\phi\cos\kappa + \cos\omega\sin\kappa \\
r_{xz} &= -\cos\omega\sin\phi\cos\kappa + \sin\omega\sin\kappa \\
r_{yx} &= -\cos\phi\cos\kappa \\
r_{yy} &= -\sin\omega\sin\phi\sin\kappa + \cos\omega\cos\kappa \\
r_{yz} &= \cos\omega\sin\phi\sin\kappa + \sin\omega\cos\kappa \\
r_{zx} &= \sin\phi \\
r_{zy} &= -\sin\omega\cos\phi \\
r_{zz} &= \cos\omega\cos\phi
\end{aligned}$$

This rotation is followed by a translation of vector $p0$, related to the world coordinates of the focal point of the camera, which gives the rigid body transformation $p_c = Rp_w + p0$. The rays equation can thus be written $p = tRp_w + p0$. The rigid body transformation can be developed:

$$\begin{aligned}
x_c &= R_{xx}x_w + R_{xy}y_w + R_{xz}z_w + p0_x \\
y_c &= R_{yx}x_w + R_{yy}y_w + R_{zy}z_w + p0_y \\
z_c &= R_{zx}x_w + R_{zy}y_w + R_{zz}z_w + p0_z
\end{aligned} \tag{5.2}$$

Though the positions (x_c, y_c, z_c) of the points in the coordinate system of the camera are unknown, the projections of these points onto the image plane can be calculated using the equations for perspective projection:

$$\frac{x_i}{f} = \frac{x_c}{z_c} \tag{5.3}$$

$$\frac{y_i}{f} = \frac{y_c}{z_c} \tag{5.4}$$

where f is the focal length of the camera, i.e. the distance between the center of the projection defined by the camera and the image plane. If we combine these equations with the previous system (5.2), we obtain two equations relating image plane coordinates to world coordinates:

$$\frac{x_i}{f} = \frac{r_{xx}x_w + r_{xy}y_w + r_{xz}z_w + p0_x}{r_{zx}x_w + r_{zy}y_w + r_{zz}z_w + p0_z} \tag{5.5}$$

$$\frac{y_i}{f} = \frac{r_{yx}x_w + r_{yy}y_w + r_{yz}z_w + p0_y}{r_{zx}x_w + r_{zy}y_w + r_{zz}z_w + p0_z} \tag{5.6}$$

These two equations are the rays equations. They are used in two cases, following which term is known or not. Their first utility is to determine the parameters R, $p0$ and f, which requires to have *calibration points*, i.e. points for which we know both world and image coordinates. When these parameters are determined, we can use them with points where only the image coordinates are known, and hence determine the rays in the world coordinates.

For the first calibration purpose, we have 13 unknowns: 9 for the rotation matrix, 3 for the translation vector and one for the focal length. However, the rotation matrix R is defined from only three angles (i.e. pitch, yaw and roll), which reduces the number of unknowns to 7. Hence, four calibration points are needed. In practice, much more points will be used to provide an accurate solution. The system can be solved using classic numerical methods, like QR decomposition.

5.3 Affine calibration

Several more or less advanced methods exist, each taking more or less camera artifacts into account. We used a simple but efficient affine calibration which allows us to determine the equations of the rays from the image coordinates, taking the following additional intrinsic effects into account:

- skew, due to non-orthogonal axes on the image plane,
- differential scaling, due to nonsquare, rectangular pixels.

The lens distorsion will not be considered here. It is much more complex, non-linear and must be determined with an iterative method.

The difference with the previous process is that we add an affine transformation of the original image plane that models distortions and errors using intrinsic parameters. If (x_i, y_i) are the measured (uncorrected) image plane coordinates and (x_{ic}, y_{ic}) the corrected ones, the affine transformation can be written:

$$x_i = a_{xx}x_{ic} + a_{xy}y_{ic} + b_x \tag{5.7}$$

$$y_i = a_{yx}x_{ic} + a_{yy}y_{ic} + b_y \tag{5.8}$$

Replacing the values of x_{ic} and y_{ic} by the values given by the perspective equations

$$\frac{x_{ic}}{f} = \frac{x_c}{z_c} \tag{5.9}$$

$$\frac{y_{ic}}{f} = \frac{y_c}{z_c} \tag{5.10}$$

we obtain

$$\frac{x_i}{f} = a_{xx}\frac{x_c}{z_c} + a_{xy}\frac{y_c}{z_c} + \frac{b_x}{f} \tag{5.11}$$

$$\frac{y_i}{f} = a_{yx}\frac{x_c}{z_c} + a_{yy}\frac{y_c}{z_c} + \frac{b_y}{f} \tag{5.12}$$

or

$$x_i = a_{xx}f\frac{x_c}{z_c} + a_{xy}f\frac{y_c}{z_c} + b_x \tag{5.13}$$

$$y_i = a_{yx}f\frac{x_c}{z_c} + a_{yy}f\frac{y_c}{z_c} + b_y \tag{5.14}$$

This formulation shows that the focal length can be integrated as a scalar multiplier of the a matrix, because it plays the role of a scaling factor. Including these equations in the rigid body transformation, we find two expressions of the form

$$x_i = \frac{s_{xx}x_w + s_{xy}y_w + s_{xz}z_w + t_x}{s_{zx}x_w + s_{zy}y_w + s_{zz}z_w + t_z} \tag{5.15}$$

$$y_i = \frac{s_{yx}x_w + s_{yy}y_w + s_{yz}z_w + t_y}{s_{zx}x_w + s_{zy}y_w + s_{zz}z_w + t_z} \tag{5.16}$$

These expressions are similar to (5.5) and (5.6), which did not used the affine transformation of the image plane. However, the s matrix is no more orthogonal, and the number of unknowns is thus 12, yielding a minimal number of calibration points of 6. The focal length has been integrated in the affine parameters a_x and a_y, as stated before.

5.4 Implementation

We implement the calibration as a linear system, solved by a classic QR decomposition (least mean square principle). For simplicity with the use of Matlab built-in functions, we used the homogeneous coordinates. Equations (5.15) and (5.15) can then be expressed as a linear system:

$$\begin{pmatrix} x_c f \\ y_c f \\ z_c f \\ f \end{pmatrix} = P \begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix} \tag{5.17}$$

where P is the homogeneous projection matrix which combines both the s matrix and the t vector. The solution of the calibration problem, i.e. the P matrix, can be found as stated above if one has a set of calibration points. The world referential in which all three-dimensional points will be computed later is the referential in which the world coordinates of calibration points are expressed.

Since we dispose of both image and world coordinates for the calibration points we can group all known terms together which yields a simple estimation of the transformation matrix:

$$\frac{P}{f} = \begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix} \begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix}^{-1} = \begin{pmatrix} x_i \\ y_i \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix}^{-1} = v_i v_w^{-1} \tag{5.18}$$

where the z_c coordinate has been replaced by its unitary value. With n calibration points, the v_i and v_w augmented vectors become matrices with a $[4 \times n]$ size. The problem is consistent since $[4 \times n]/[4 \times n] = [4 \times 4]$.

The z_c value is not revealing in our problem, since the image plane is always at location $z_c = 1$ in the camera referential so that the focal length will be expressed in the world referential units (e.g. millimeters). It will be used below when checking the validity of the calibration algorithms.

5.5 Validation of the calibration method

Because the whole process relies on calibration, we want to be sure that it has been conducted correctly. Two methods were used: checking for the z_c coordinate which must be equal to 1, and verifying that the camera location is correct.

5.5.1 Camera location

Our approach is to compute a bunch a rays and locate their intersection point. We therefore use the calibration data, i.e. the image coordinates of calibration points, from which we can construct the associated rays, which should (nearly) intersect at the focal point location.

	X [mm]	Y [mm]	Z [mm]
Calibration 5, left	-363.59	243.41	1227.1
Calibration 5, right	894.18	238.11	1254.1
Calibration 6, left	-877.18	226.44	1245.9
Calibration 6, right	1191.4	246.17	1267.1

Table 5.1: The camera locations according to the projection matrices for the two calibration setups made in February, known as calibration 5 and 6.

Table 5.1 presents the different locations of the cameras for calibration 5 and 6. The resulting positions match the experimental setup. For example, the cameras were placed roughly in the middle of the 500mm-wide flume, and the Y locations are indeed close to 250mm.

5.5.2 Unitary z_c coordinate

We check for the unitary of the z_c coordinate by simply looking if the two last lines of the projection matrix P are equal. This check does not depend on the data but rather on the algorithm used and the definition of the P matrix: the two last lines will always be identical if the algorithm is correct. The identity has been confirmed for the algorithm, hence for all calibrations.

5.6 Limitations of the affine calibration

The calibration process explained above showed a serious limitation when calibrating the upper camera (the high-speed camera, used for velocimetry). This is because the camera is nearly aligned with the Z axis of the world coordinate system, so that the disparities of the calibration points for the different heights are small. Combined with its limited resolution, the input data for the calibration process was not revealing.

The very little corrections together with these great errors yield an ill conditioned problem when inverting matrices in the calibration process.

Because the velocimetric process does not take the depth into account, inaccuracies in the depth estimation is not important, so it could be used to rectify images. However, in the case of a stereometric analysis, the depth accuracy is capital and using the upper camera will not yield significant improvements, as we observed.

This is frustrating because efforts had been made to be able to combine three cameras for a better triangulation and particle tracking. The upper camera has thus not been used in the stereometric aspect of our experiments.

5.7 The calibration setup

As stated above, the calibration process requires to have a special set of points for which both world and image coordinates are known. Since the position of the pearls on the fluid are not known, these images contain no useful information for calibration, and we must proceed through a special setup to get the calibration points. We describe two calibration setups, the first used for the November experiments, the second, much more precise, designed for the final February runs.

5.7.1 A first calibration attempt

The affine calibration requires at least six calibration points for determining all of its parameters. We thus placed a concrete cube, on a flat surface over the sediment bed, its corners being the calibration points (fig. 5.1). Because the right camera was vertical, the cube was placed on one of its edges so that six points were visible. This calibration has been very badly conducted for several reasons. One should first remember that the world referential in which the calibration points are expressed will be used to express the three-dimensional coordinates obtained by the whole process. For obvious practical reasons, we wish to have a world referential with its axes parallel to the principal axes of the flow, which are also the principal axes of the flume. In the case of the cube, we attached a referential to three concurrent edges. Because the angles made by the edges with the flumes principal axes are not known, it is not possible to define a transformation matrix to pass from the cube referential to another referential orthogonal to the flume.

The transformation from one referential to the other was made by calculating the

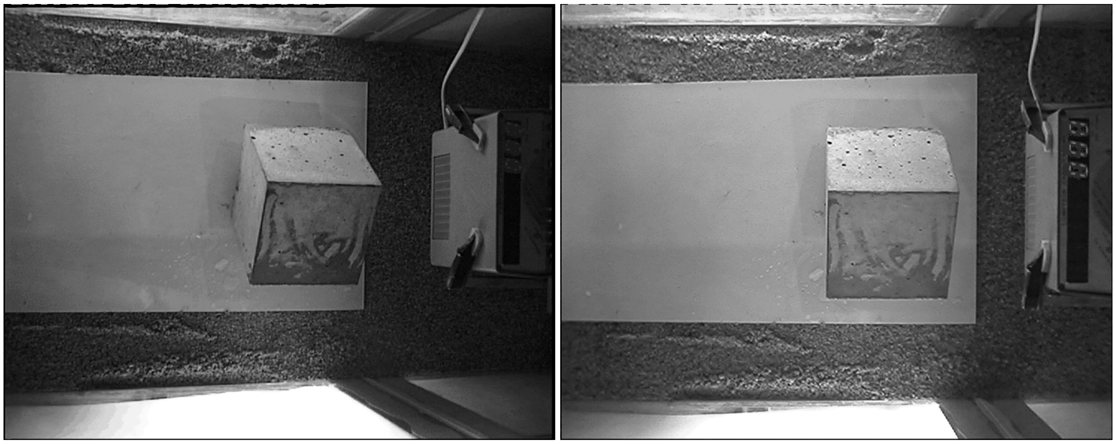


Figure 5.1: The left and right calibration shots for the first setup.

equation of the sediment bed in the cube referential (using feature points present on the sediment bed), then impose its equation in the flume referential, and at last compare those equations to define the transformation. This complex process gave us a transformation matrix for a referential nearly parallel to the flume's axes, but in which the boundaries of the flume were not known, making it impossible to define the limits of the flume. Also, the calibration involved only six calibration points, while we should use many more to have a stable and precise camera location. Another problem is that all the points of the cube are located in a small region of the image plane, leading to small distances between them and thus data of a poor quality.

5.7.2 A convenient calibration setup

The problem usually related to a good calibration setup is to be able to have many calibration points, for different depths. This cannot be done with a solid opaque object because we cannot define calibration points inside the object. One could then think of using a wire frame object, for in that case inner points are visible and can be used. However, wire frame objects are difficult to build and to preserve from deformations. This brought us to the point of defining a new system for calibration.

While all systems presented above take all informations in one shot, our approach is to take several shots of the calibration scene, changing the object position between shots. This is convenient for static cameras, but cannot be used for moving camera methods. The moving object was a plane, fitted with regularly spaced calibration tags, and the plane could be precisely positioned along the Z axis of the flume with 4 screws. A view of the calibration setup is presented on fig. 5.2, while typical

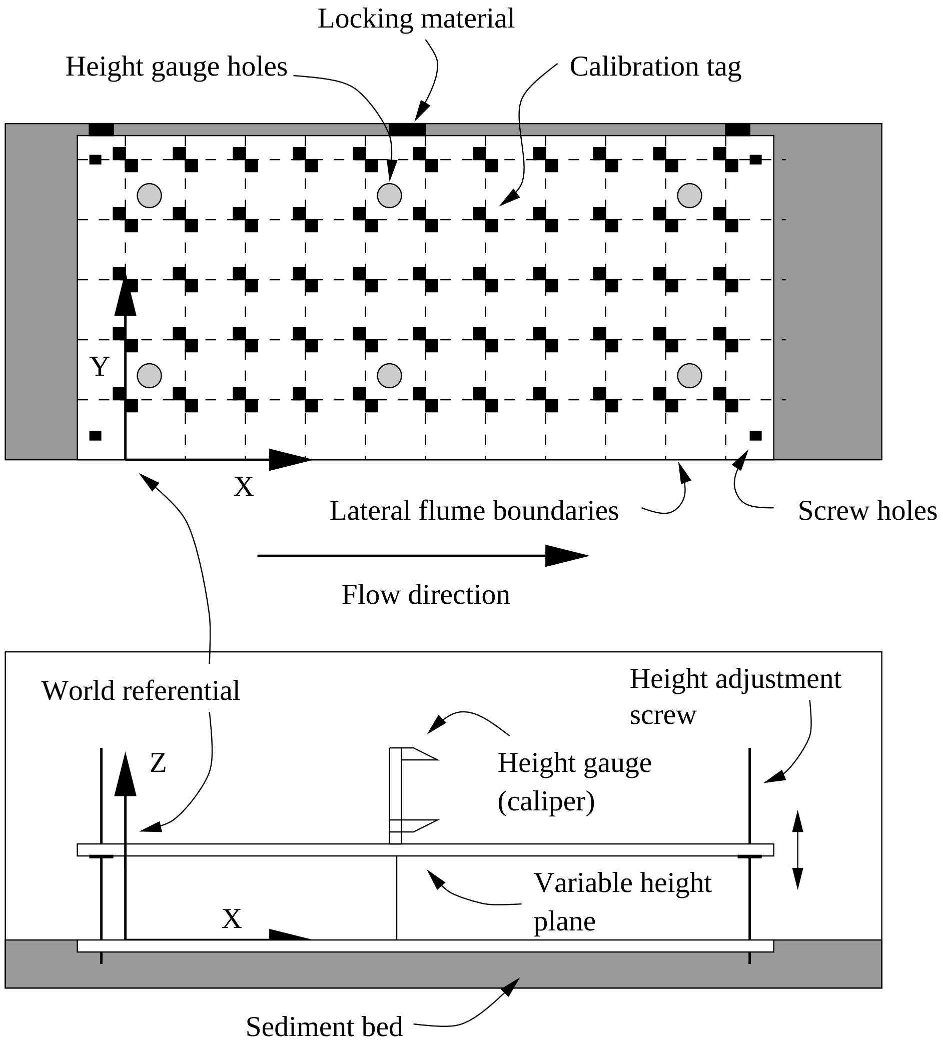


Figure 5.2: The calibration setup.

calibration images are shown on fig. 5.3.

Calibration shots were made for four different Z altitudes: 50mm, 100mm, 150mm and 200mm. The number of calibration points then reaches the reasonable amount of 130.

Two calibration setups were conducted, involving different camera positions. For the first (calibration 5), used for runs 3 to 5, cameras were placed with angles such as to have a reasonable margin before occlusions. The last run 6 involved a calibration with more spaced cameras (calibration 6), which also means smaller angle since we want to observe the same portion of the flume. In this case, the angles involved were the lower limit to avoid occlusions. Cameras positions for calibration 5 and 6, together with the positions of the cameras for the November runs (calibration 1), are

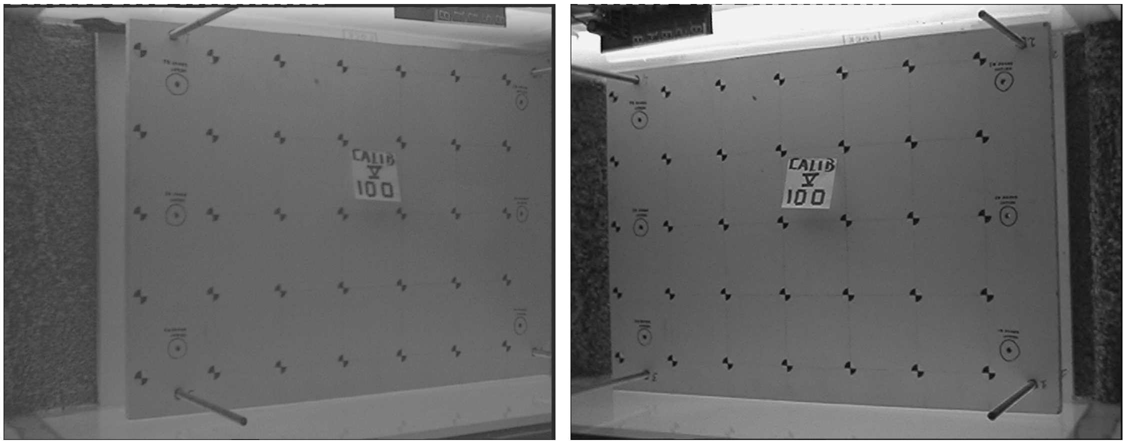


Figure 5.3: A calibration shot for each camera, at a plane altitude of 100mm.

shown on fig. 5.4.

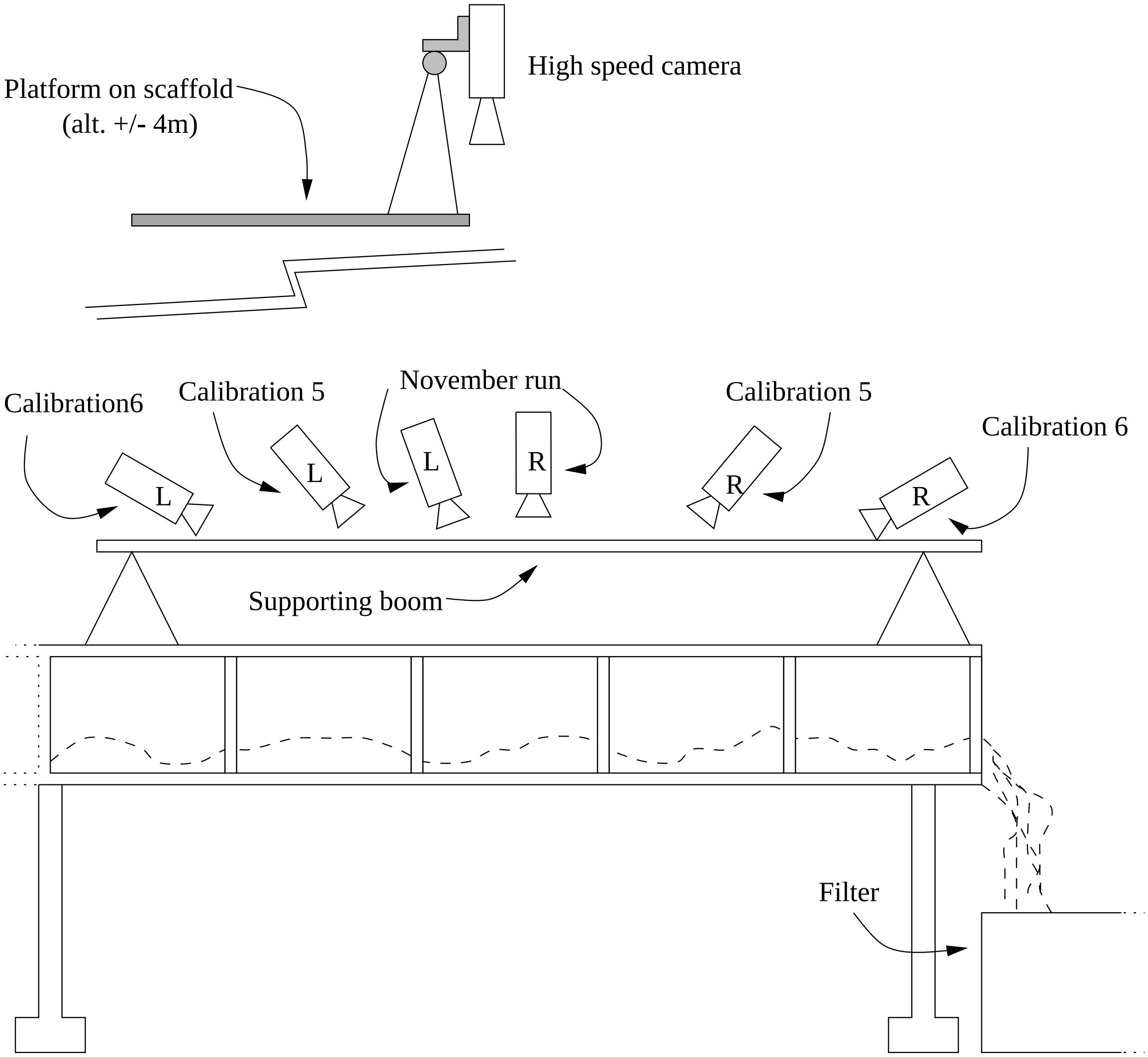


Figure 5.4: The different camera positions for the three calibration setups. The high-speed camera was always located at the same place.

Chapter 6

Particle detection

6.1 Introduction

The approach for this first step is driven by three main goals: we want to detect particles and only particles, accurately, and in an efficient way. The latter aspect is important here because we work with high resolution raw images, which represent a considerable amount of data to process. Such a raw image is presented on fig. 6.1 The process shown on fig. 6.2 consists in transforming the original image with a variation of a wavelet transform presented by Strickland & al. in [9] and modified for our purposes. We then look for the peaks in the transform which represent the particles and at last eliminate as much parasitic detections as possible. Because of the importance of the wavelet transform in the process, we will first briefly describe the associated theory. If the reader is unfamiliar with this transform, we recommend reading [10] and [11].

6.2 The wavelet transform

The undecimated two-dimensional wavelet transform (fig. 6.3) consists in filtering an original image with two unidimensional filters, a high pass (G) and a low pass (H), called wavelets. Since the filters are unidimensional, we can apply each filter vertically or horizontally. This yields four different output images given the two possibilities for filters and directions. We call these four transformed images d_{HH}, d_{HL}, d_{LH} and s_{LL}. The d_{HH} image contains the details which appear in both vertical and horizontal directions, d_{LH} and d_{HL} both show details in one direction, the last one (s_{LL}) being low-passed. The s_{LL} image will not be used here since it is only a smoothed version of the original image, containing no interesting information in our case. It is used for further decomposition in multiresolution processes. Fig. 6.4 shows the different

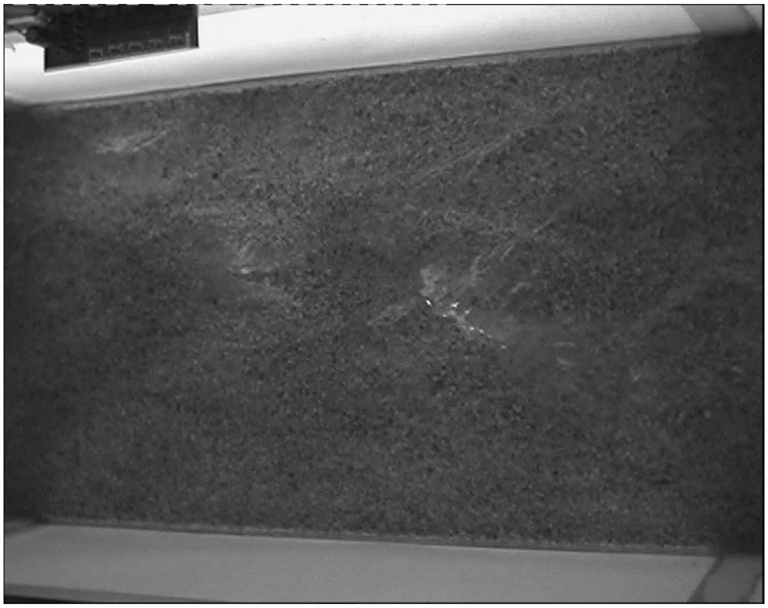


Figure 6.1: A raw image from the right camera during run 4.

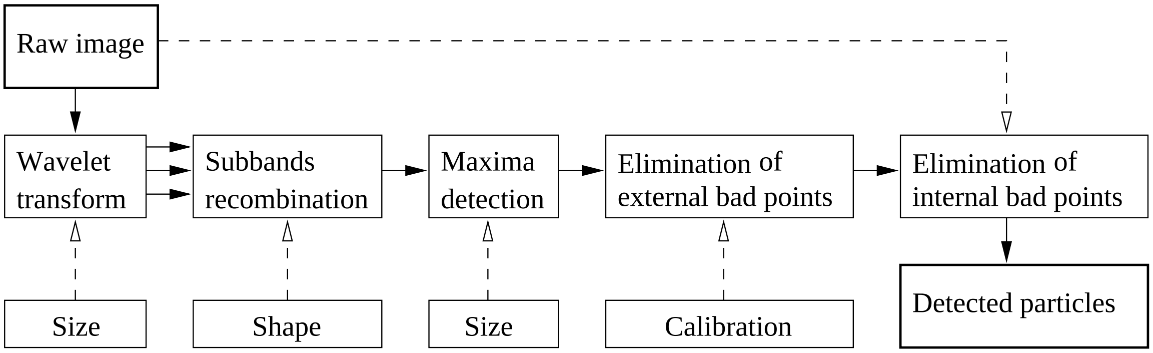


Figure 6.2: The detection process. Dashed lines indicate parameters influence, continuous lines the data path.

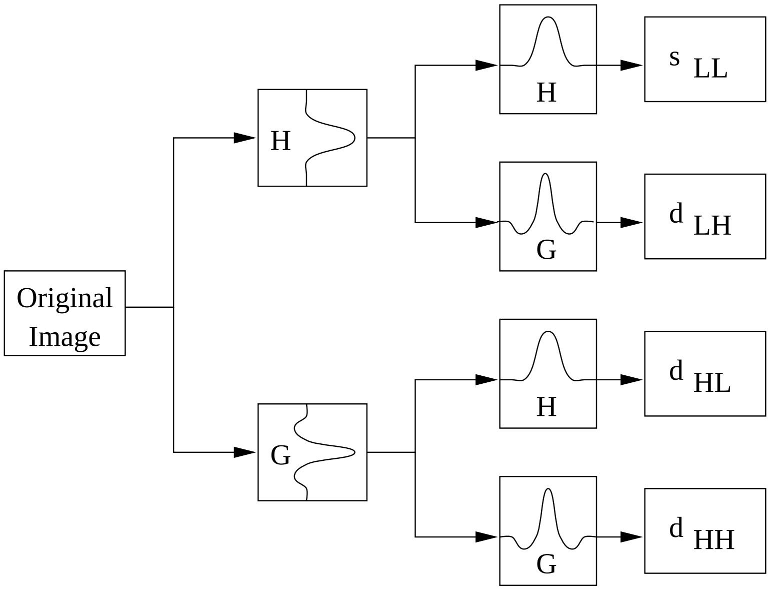


Figure 6.3: The filter based on a two-dimensional wavelet transform.

outputs images. The undecimated aspect of the transform is important here to keep a point-to-point correspondence between the transformed images, and so recombine them easily.

6.2.1 Matched filters

The H and G wavelet filters were adapted to the size of the pearls, their global shape and the surrounding noise. Strickland & al. showed that the wavelet transform could approach an adapted filter to detect gaussian objects in a Markov background noise if the wavelets were of the form:

$$H(n) = \gamma(n) \tag{6.1}$$

$$G(n) = -\frac{\partial^2}{\partial n^2}\gamma(n) \tag{6.2}$$

where $\gamma(n) = e^{-\frac{n^2}{2\sigma^2}}$, and σ depends on the size of the objects to be detected. In our case, particles can reasonably be considered as gaussian objects, while fig. 6.1 shows the water flow and sediment background, closely matching a Markov model.

The size of the filter was chosen so that the width of the central positive peak of G is the diameter of the pearls, which ensures optimal detection. This width is approximatively equal to 2σ over a wide range, and since the pearl diameters are between eight and twelve pixels, we choose sigma equal to 4. The use of the lower

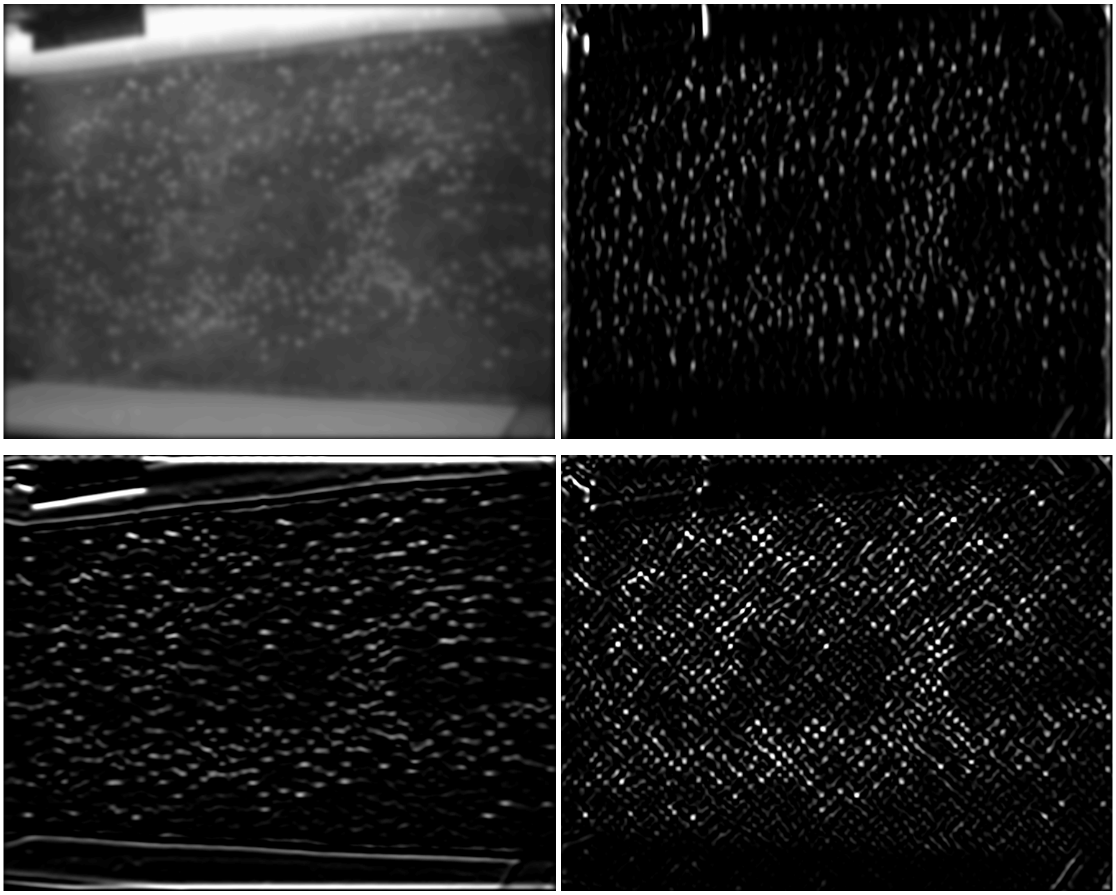


Figure 6.4: The four wavelet transforms: upper left: s_{LL}, upper right: d_{LH}, lower left: d_{HL} and lower right: d_{HH}.

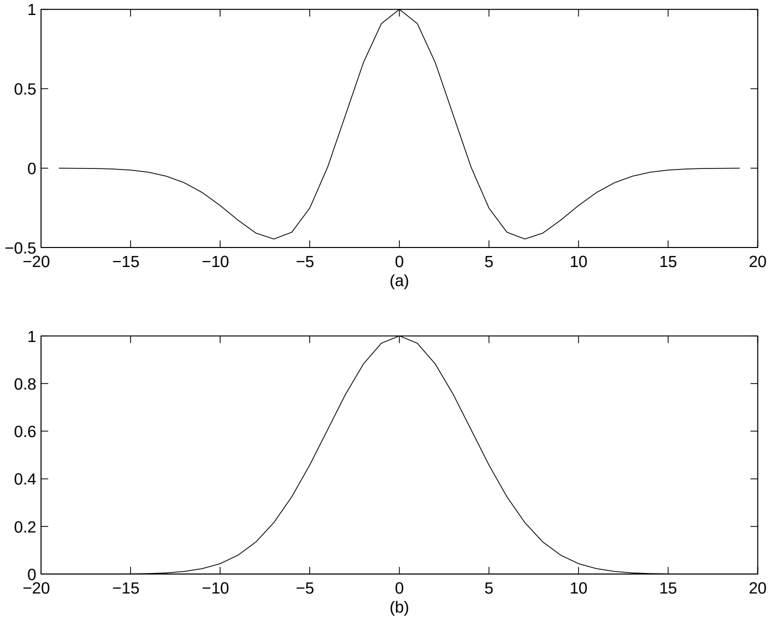


Figure 6.5: The two wavelet filters: (a) high pass and (b) low pass.

value for sigma is due to the fact that smaller pearls are also dimmer and thus need a well adapted width for the filter, while large pearls also appear brighter and can afford a approximative width value. With this value, the length of the filter can be limited to 21 without degraded performances. The resulting H and G filters are shown on fig. 6.5.

6.2.2 Subband recombination

We now wish to combine the three useful output images properly, so that only pearls finally appears on the resulting image. Since pearls are circular, they have the same vertical and horizontal properties, hence the same response to the filters in both directions. This property, related to their shape, make the d_{HH} image attractive, but one can see on fig. 6.4 that the d_{HH} image contains much more noise (it is high pass) and is thus not suitable alone. On the contrary, d_{HL} and d_{LH} images isolate particles without too much background noise, but are independently insufficient, for the pearls are precisely located in only one direction and these images have also located the parasitic linear boundaries of the flume.

Because the centers of particles are present on both images and parasitic boundaries are not (one image contains the verticals, the other the horizontals), we could apply a kind of AND operator between d_{HL} and d_{LH} images, for instance a point-to-point product. This will both reinforce the particles locations and dimm the parasitic linear features. The $d_{LH}.d_{HL}$ and d_{HH} pictures are not equivalent, however: the $d_{LH}.d_{HL}$ is less noisy, due to the low-pass filtering.

Since the d_{HH} image still contains useful information, we try to incorporate it in the $d_{LH}.d_{HL}$ image. We do this by first rescaling the $d_{LH}.d_{HL}$ image, using a square root, then adding the d_{HH} image:

$$T = \frac{1}{2}\left(\sqrt{d_{HL}.d_{LH}} + d_{HH}\right) \tag{6.3}$$

Negative points of the transform are simply zeroed before applying the square-root to avoid complex numbers. Since negative points correspond to the worse points, they can be ignored without degrading the performances. This recombination process is not very formal, but it yields good results for the final transform T (fig. 6.6). Also, we did not use the multiresolution aspect of the wavelet transform, because all particles are of the same size. It is thus more interesting for both detection quality and computing time to adapt the filters directly to the particle size. Particularly, because of the undecimated aspect of the transform, a multiresolution approach would present much more time and memory requirements. The fact that the resulting images are redundant from the point of view of information quantity because the process is undecimated is not pertinent here.

We have also tested a multiresolution Haar wavelet transform, but the quality was poor for it is not adapted as the transform we use. It also shows that multiresolution does not improve results here in a significant way if its filters are not adapted. The multiresolution transform presented in [9] was also tested, but it was much too slow.

6.3 Maxima detection

Now that we have a transformed image, we need to look for maximas in order to detect particles. Simply looking for maximas and eliminating it for further search would not be sufficient because points near the maxima corresponding to the same pearl also have high values that will be detected as distinct maximas later. To avoid such problem, we clear not only the detected maxima but also the points surrounding it. Such clearing of the whole peak prevents the detection of false maximas.

Practical implementation is done by zeroing all points near a detected maxima. The clearing radius must now be determined for optimal detection. Since the size of

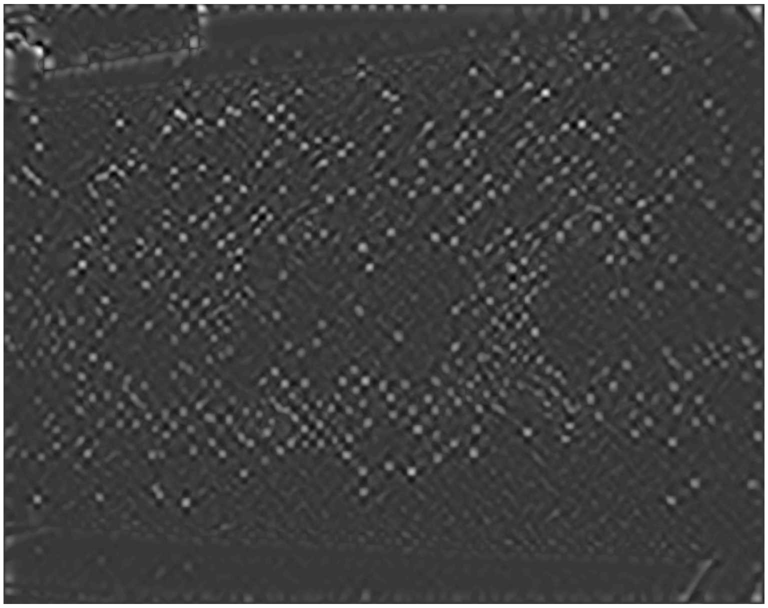


Figure 6.6: The final recombined transformed image.

the pearls in the transformed image are the same as the pearls in the original image, the clearing radius can be defined as the radius of pearls, i.e. 6 pixels. We prefer to use the largest radius of the pearls because clearing more is better than an insufficient zeroing. Indeed, if the clearing radius is too small (fig.6.7d), parasitic particles will appear around the correctly located particle, while clearing more will relocate or even completely mask the second particle (fig.6.7c).

The detection of parasitic particles is also diminished by including a derivative condition: a maxima is selected only if the local derivative is small in the original transformed image.

To limit the search of points (which is time consuming), we have set a threshold in the process. It consists in zeroing all points in the transform image that are inferior to a certain percentage of the global maxima of the that image. The search will be made until a certain number of points is reached or the transformed image is all-zero, which mans the zeroing dishes have flatten the whole transform to zero. This latter point prevents the detection of lots of wrong particles if very few real particles are present (for instance at the start of the runs, where the particles have not yet covered the entire observed surface).

The maximum number of particles was set to 500, for the filmed portion of the flume is more or less 1m long, while we spread 1000 pearls on a 2 m long grid. Hence, when the particles cover the whole surface in view, their number is close to 500.

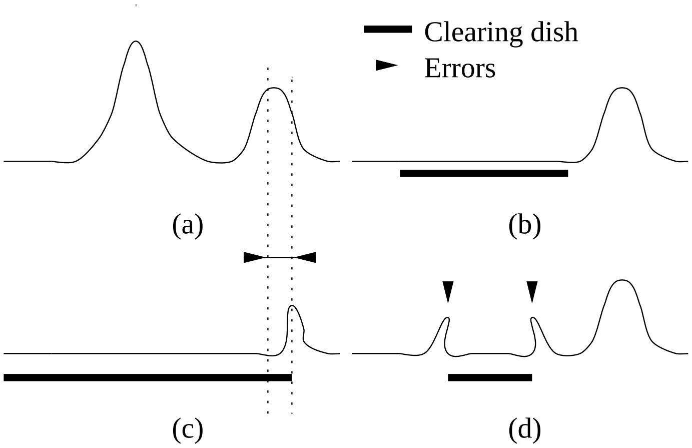


Figure 6.7: A scheme describing the influence of the clearing radius on the detection of parasitic peaks: (a) original peaks, (b) a correct clearing, (c) relocation of the second particle due to a too large clearing radius and (d) parasitic peaks that occur if the radius is too small.

A too high zeroing level will cause the algorithm not to detect some particles, on the other hand, a lower level will lead to a more numerous detection of parasitic points. A correct compromise was found to be around .05, i.e. zeroing all point below 5 percent of the global transformed image maxima.

Another aspect is the precision due to the limited resolution of the image. It can be improved using an interpolation algorithm to reach subpixel accuracy: once a maximum is found, we interpolate its position in the 9×9 pixel grid around it. Because of other artifacts presented before, the gain of such interpolation is not very significant (see Chapter 10), but it was still implemented to keep a general tool for stereovision.

6.4 Detection artifacts

Some unwanted detection artifacts appear. We can point the fact that the maxima in the wavelet transform will maybe not be the position of the particle center. This can be due to the facts that the maximum of intensity in the original image does not occur on the particle center because of lateral lighting. Fig. 6.8 shows that some particles have a quarter moon-like shape, instead of circular dishes. This aspect has not been taken into accuracy, but does not seem to influence the process in a significant way (it is probably cancelled by the low-pass filtering present in the wavelet transform). A possible improvement would be to integrate the theory presented in [17].

Another artifact is the non-zero time of exposure, which causes the particle to be

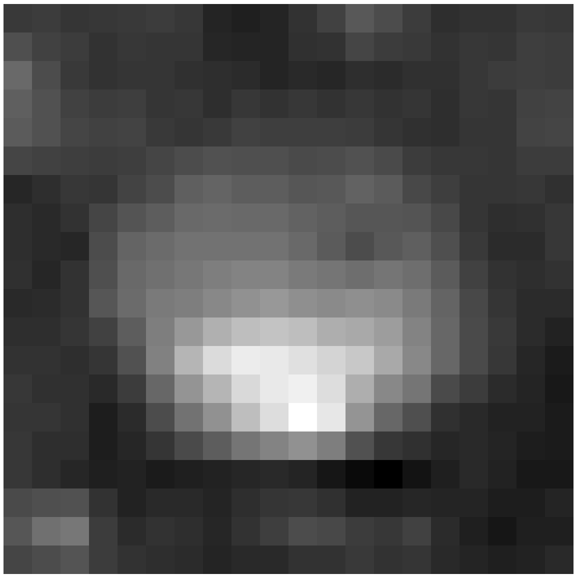


Figure 6.8: A close view of the shape of a pearl showing the dissymmetry in the original image intensity.

oblong. As yet, it has been neglected for it is not visible on the pictures.

6.5 Elimination of external parasitic points

Unwanted parasitic points are detected on the flume, the counter,... These points, which are external to the fluid part of the scene, create parasitic boundaries effects when using the Voronoï interframe matching (see section 7.3.1). We eliminate these bad points using the calibration data.

Each point coordinates is projected on a horizontal plane in the world coordinates system, thus a plane parallel to the flume bottom. The altitude of the projection plane is chosen to be approximatively the midplane of the fluid surface, i.e. 25mm. Projected points with Y coordinates less than 0mm or larger than 500mm are rejected, on the basis that they belong to the flume armatures and the counter. Fig. 6.9 and fig. 6.10 show the detected particles before and after that process.

In the second experiments, white particles sometimes reflected on the glass windows of the flume, when near to it. Since it happened on both left and right views, and for the same particles, the matching algorithm considered these to be real particles, not artifacts. This external elimination is also a convenient way to avoid such bad matches.

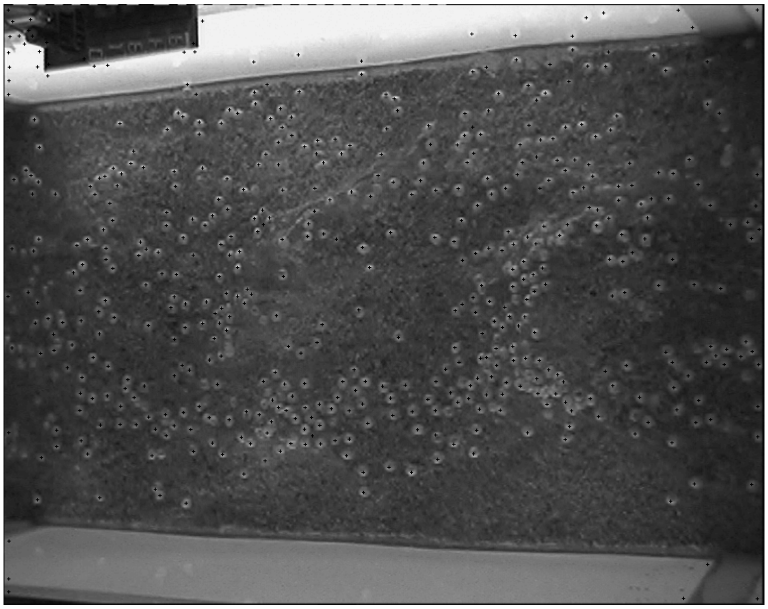


Figure 6.9: Detected particles, indicated by black crosses.

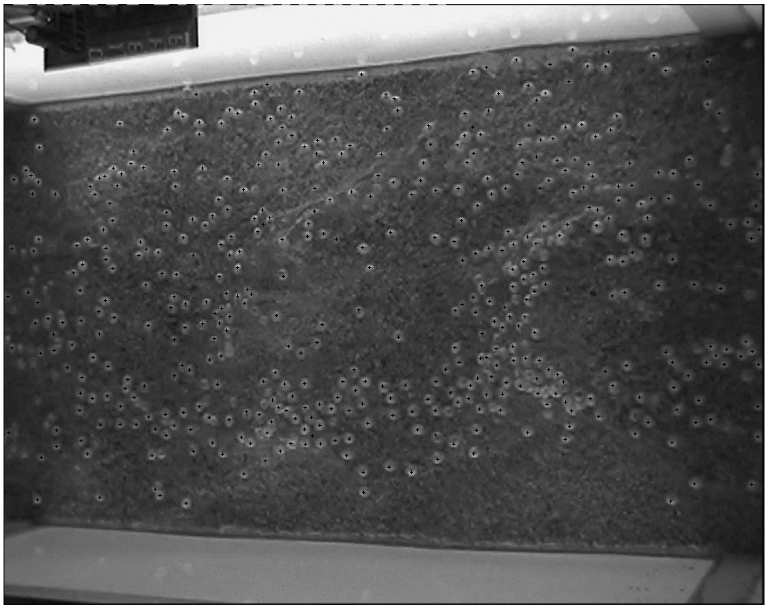


Figure 6.10: After the elimination of false particles on the flume armatures, only points in the fluid part of the scene remain.

6.6 Elimination of internal parasitic points

Parasitic points that are internal to the flume boundaries cannot be eliminated so easily. The elimination of those points requires to find a characteristic specific to particles compared to the background. This will not be their size, for it has already been treated by the wavelet transform.

Two special characteristics can be defined, both based on the contents of the original image. This idea in fact comes from the multiresolution aspect of the wavelet transform: one can see the original image as a 0-level wavelet transform. The use of a 0-level transform and a higher level transform is thus in some way multiresolution.

The two characteristics are also close to the idea of wavelets. These are the average and variance of the neighbourhood of the detected point. We can see the similarity between these notions and respectively the H and G filters of the wavelet transform. The advantage is that checking the intensity and variance at selected locations is faster than computing a whole transform.

The use of the average is well suited for the first experiments, where the particles are black on a clear background (white particles are considered as parasites). Because the average light level of a particle is different from the background, we compute the average of the neighbourhood of the point, and fix a threshold above which we consider the point to be a particle. Fig. 6.11 and fig. 6.12 illustrate the considerable gain of efficiency for these experiments.

The problem with the second set of experiments is that the particles where white on a clear background. The use of intensity would thus provide poor improvements, since both particles and non-particles have the same intensity characteristics. We can then use another statistic property of the neighbourhood: its variance. Particles are nearly uniformly lit objects, thus yielding a relatively small variance. On the contrary, bad particles consists on a noisy background and its variance is thus expected to be high compared to good particles. The results of this method based on the variance were disappointing for unidentified reasons.

6.7 Conclusions

The detection process exposed above is efficient, for 95 % of the particles shown on fig. 6.10 are detected, while the detected points consist in 99% of particles.

However, the price to pay is high and processing the 50 image pairs of a run

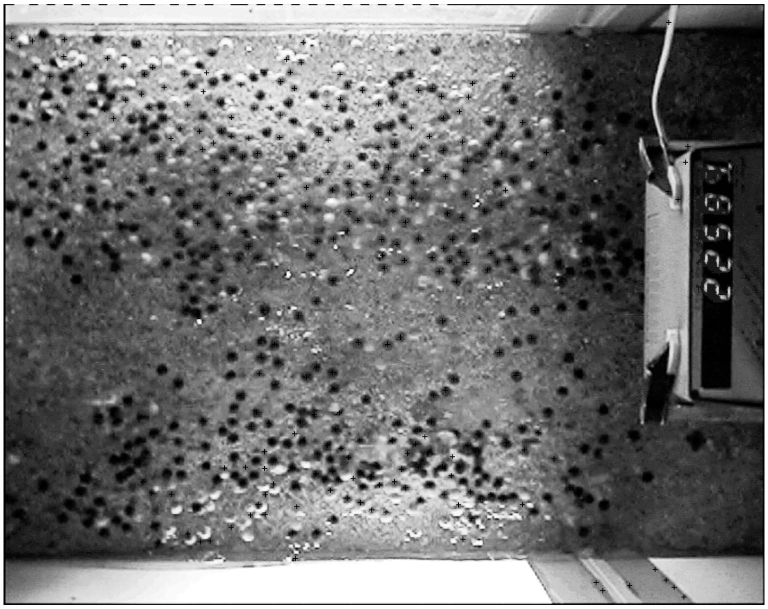


Figure 6.11: Parasitic detected particles for run 0. Bad detections of white particles are visible in the highly lit zones of the image.

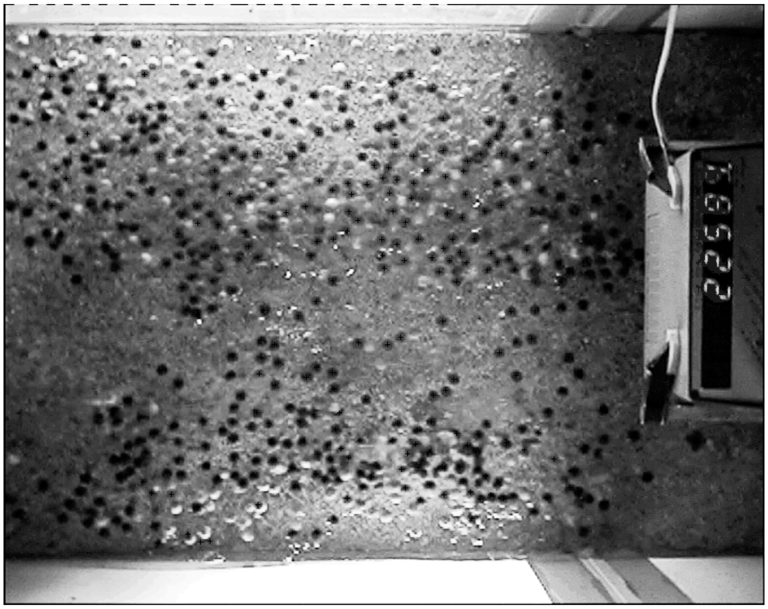


Figure 6.12: Parasitic detected particles after an average light level check: most of the white particles are no more detected.

(100 images in total) takes several hours. A possible improvement for this method would be to introduce mathematical morphology (see [22]) for the detection stage, in replacement for our clearing-dishes algorithm. Another way to detect particles is also presented in [7].

The available data is now the list of detected points for each image which, together with the calibration matrices, represent all the necessary information required for matching and depth estimation.

Chapter 7

Matching and depth

Once the calibration process is conducted, we are able to compute the equations of the rays in world coordinates, with the use of the two projection matrices P_r and P_l. In an ideal case, those would intersect at the location of the particle in the world coordinate system, if and only if the two rays correspond to the same particle. However, in practice, the rays will not intersect exactly due to errors and the limited accuracy of the detection process, of the calibration and the over-determined nature of the problem. The position of the particle in the world coordinates system has been estimated as the midpoint of the segment orthogonal to both rays, or equivalently the average of the coordinates of the two points of the rays closest to each other (fig. 7.1). This is computed as follows. If we express the rays in their parametric form

$$R_r = r_r + k_r v_r \tag{7.1}$$

$$R_l = r_l + k_l v_l \tag{7.2}$$

we can determine the two k paramters corresponding to the closests points on the rays by solving the linear system:

$$\begin{pmatrix} v_r v_r^T & -v_r v_l^T \\ v_r v_l^T & -v_l v_l^T \end{pmatrix} \begin{pmatrix} k_r \\ k_l \end{pmatrix} = - \begin{pmatrix} (r_r - r_l) v_r^T \\ (r_r - r_l) v_l^T \end{pmatrix} \tag{7.3}$$

These expressions come from the fact that the two directing vectors v_r and v_l of the rays are orthogonal to the segment on which the reconstructed point is to be found, hence their dot product must be zero. When the two parameters are known, the reconstructed point for the three-dimensional point P can be easily found by averaging the closest points P_r and P_l on the two rays:

$$P_r = r_r + k_r v_r \tag{7.4}$$

$$P_l = r_l + k_l v_l \tag{7.5}$$

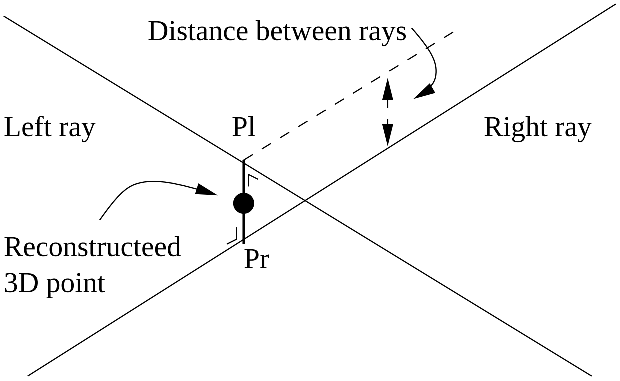


Figure 7.1: The matching geometry.

$$P \approx \frac{P_r + P_l}{2} \tag{7.6}$$

Now that we saw that the depth could be easily computed once when we had a pair of matched points, the last problem we have to face is the matching.

The intersection of the rays requires to have two (left and right) image coordinates corresponding to *one* detected particle. We used several different methods to determine the matches, but all of these involved the same matching criteria: the distance $||P_r - P_l||$ between the rays of the particles to match had to be minimal.

The selection of matches is done with the following algorithm, based on an affinity matrix. Such matrix is defined as

$$A(i,j) = \text{affinity between points } i \text{ and } j. \tag{7.7}$$

The affinity property can be a distance between the i and j points, or a more complex matching likehood, like the neighbourhoods (see below).

We first search for the global maxima of the matrix, we then exclude the (i,j) couple found for that maxima from further search. This is done by zeroing the i raw and the j column (or, if values in the matrix can be negative, by replacing the i raw and j column by $-\infty$ values). We then search the next maxima of the matrix,...

This method is based on the fact that we use it when matching points. Indeed, the matches are exclusive, i.e. a particle in one image plane cannot be matched with more than one on the other image plane, which forces us to exclude the two image coordinates of those particles for further matches.

A possible improvement would be to add another property like the distribution of the affinities of the line and column corresponding to a matched pair. One could then check if the distribution is very sharp, or if all values in the line/column are nearly the same, which would mean a rather noised match we could not take into account.

7.1 Midplane projection

In this first attempt we used the fact that the vertical projection of the left and right particles field onto a Z-plane corresponding to the midlevel of the fluid surface will be nearly the same. It should be noted, however, that they will never match exactly since this would mean that the fluid surface is flat, which is not the case. In fact, changes in particle's altitude will lead to relative displacements on the projection plane. The matching criteria is simply here the distance between point in the projections. If the cameras are close enough in both distance and angle, variations will be small (but depth less precise) and this matching method will work. If not, disparities will be high, and the two projections will be difficult to match.

The problem is to choose the midplane level. In fact, this level will influence the mean level of the final surface, since best matches will occur for this level: if the particle is located at a Z coordinate equal to the chosen plan altitude, the distance between its representations will be nearly zero and the a match will be made. For other heights, the distance will grow and the matching capability will progressively decay, leading to less and less matches. The optimal planes for the different runs are displayed on table 7.1. They roughly correspond to the midplane of the water surface. This process was used in the first experiments, were distance and angles of the cameras were close. In the other experiments, the angle and distance were sensibly higher (to gain in precision), and other methods were developed.

	Run 3	Run 4	Run 5	Run 6
Mean plane altitude [mm]	38	32	42	27

Table 7.1: The mean plane altitude for the different runs

7.2 Exhaustive matching

The idea is to compute rays distance for all possible pairs of particles. We will say that there is a match when rays pass close by. We thus compute a distance matrix,

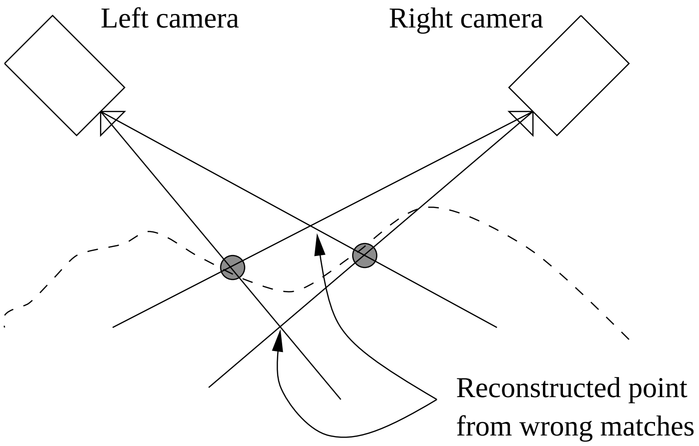


Figure 7.2: The problem of mismatches caused by a pair of particles aligned with the baseline.

then look for best match (minimal distance), the next best match,... When we found a match, we eliminate matched particle from the search list.

The problem associated with this approach is that if two particles are nearly in a plane containing the baseline, the distance between their rays in left and right images will always be small. Indeed, the epipolar line and the two points are (almost) in the same plane, hence rays always (nearly) intersect whatever the combination (four in total). This is illustrated on fig.7.2. It is possible to eliminate those mismatches by checking the Z coordinate of the points, and keep only those within a certain margin (our choice for this parameter was the 0mm-80mm interval). Indeed, if two particles locations are far from each other, a mismatch will lead to a high $|Z|$ value. However, for closely located particles, it is not possible to check the Z coordinate since both will be in the proper range. The number of points eliminated for out-of-range reasons is very small, less than 1.5 %, showing the good quality of the matching process.

This method is very attractive since it does not depend on a particular value like the previous Z-plane altitude. However, it is time consuming: if we have 500 particles, 250.000 distances between rays are to be measured.

7.3 Neighbourhood operations

This is a variation of the first method. We still project the particles fields on a midplane, so the height artifact remains. The major advancement is that the search for the matching particle is no more based on simple distance, but rather on the configuration of its neighbours. The height artifact, though still present, is diminished because we take another parameter into account: the shape of the neighbourhood. The tool we selected for analyzing the neighbourhoods is the Delaunay triangulation, together with the Voronoï representation. Extensive analysis of this method can be found in [12].

The data processed by these algorithm are two series v_1 and v_2 of 2D coordinates, in our case corresponding to detected pearls, each vectors issued from a different image plane. These image planes can be issued from two different cameras, which is our concern here, or come from a single camera but taken at different times, which will be the case in Chapter 9.

7.3.1 Delaunay triangulation

This first stage processes the two sets of points separately. For each particle, we determine the list of its neighbours. Since the neighbourhood property is reflexive, we will use straight unoriented lines between points to indicate neighbours.

The data vector v_i is processed in a single pass, by adding and withdrawing neighbours. For each point, we go through the whole v_i vector and look for possible neighbours. A point i is considered as a neighbour of the point j if the neighbourhood line ij crosses no smaller neighbourhood line issued from j. If ij crosses longer neighbourhood lines, it is removed. We then have the mesh shown on fig. 7.3.

This approach is convenient because the distance criteria are relative: in a sparse region where the points are sparsely scattered, neighbourhood lines are all long, but only the smallest of those will be kept, even if there exists much smaller distances in other parts of the scatter. In dense regions all lines are small, but only the very smallest will be kept. The process is thus independent of the particles density, but strongly relies on the particles organization.

7.3.2 The Voronoï polygons representation

Voronoï polygons are based on the Delaunay triangulation, in the way that the list of neighbours for each point is its input data. These polygons are in fact nothing more than another way to represent the neighbourhoods: while the Delaunay triangulation showed the neighbourhoods as links *between* particles, the Voronoï polygons show the neighbourhoods as polygons *circling* the particle (fig. 7.3).

The transformation for one representation to the other simply consists in replac-

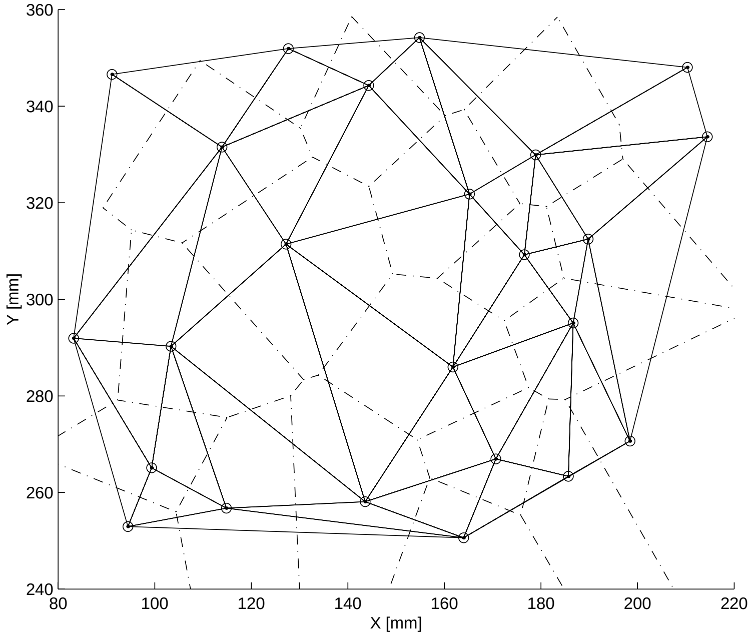


Figure 7.3: Particles (circles), Delaunay triangulation (continuous) and Voronoï polygons (dashed) of a part of an analysed scene.

ing each Delaunay line by its median. Because this approach shows results in a more intuitive way, which will be useful for qualitative interpretation, it will be used throughout this text.

7.3.3 Matching points from neighbourhoods

As yet, we have only described the neighbourhoods, but no comparison between them has been made (as stated above, we worked with the two sets v_1 and v_2 separately). We will now focus on the way to compare them, in order to match particles.

The global idea underlying the algorithms that will be described hereafter is the following. We want this approach to be faster than an exhaustive matching, hence only some pairs that are plausible matches will be compared, the selection of these pairs being made by the Delaunay triangulation. Several ways exists for the selection and comparison of plausible particles. Their efficiencies in computing time will be compared using a honeycomb approximation for the number of neighbours.

7.3.3.1 The honeycomb approximation

We want to know in which way the complexity of the search will grow as we observe more neighbours. Instead of developing a complex theory of this problem, which

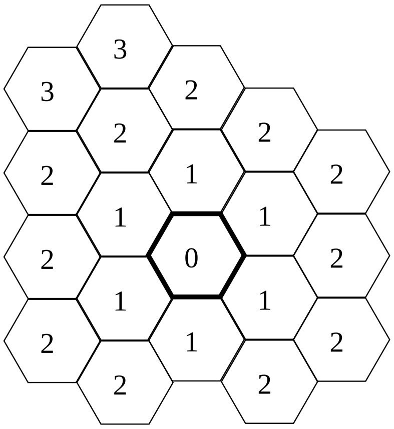


Figure 7.4: The honeycomb likehood with the Voronoï representation. Numbers indicate the odrer of neighbours

would be pointless here, we will give a simple approximation for the number of neighbours for each point. This number provides us a good idea of the operation cost of the algorithm.

The idea is to observe a classic hexagonal honeycomb mesh (fig. 7.4), and compare it with the Voronoï polygons. The strong similarity makes us think that maybe in the Voronoï mesh, a point will also be surrounded by an average of six neighbours. We computed this number for a whole set of detected particles on a single image, an found $N_{avg} = 5.924$ neighbours, which is close to 6. The difference with the ideal 6 is due to boundaries of the domain, for particles located there have an average of three neighbours, not 6. Given the high density of points inside the domain, however, their influence is small in average.

The average of 6 neighbours might indicate a Poisson-like distribution of the particles, but no further study has been conducted to include this discovery in the process.

The honeycomb approximation appears as a good theory for approximating the number of neighbours for a particle. We now define as first order neighbours the neighbours of a particle (the zero-order being the particle itself). Second order neighbours correspond to the neighbours of the neighbours of the particle, and so on. Following the honeycomb approximation, a particle will have 6 first order neighbours, 18 of the second order,... In short, the total number of neighbours will be

$$N = 1 + \sum_{k=1}^{n} 6k \tag{7.8}$$

where n is the maximal order taken into account. The first cumulated numbers of neighbours are: $N_1 = 7$,$N_2 = 19$,$N_3 = 37$ and $N_4 = 61$.

7.3.3.2 Delaunay based selection

This approach compares each particle in a first image (called *reference* image) with the closest particle and its first order neighbours in the other image ($N = 7$). This approach supposes that the effective matching point in the second image is in the neighbourhood of the particle in reference image given the first search for the closest particle. This is not a great limitation, however, because we will always suppose small disparities between the two image.

This method is very effective when disparities are small as compared to the minimal distance between particles. If the distance gets higher, we can always include the second order neighbours in the search, or even third order, with the price of a much slower algorithm. Taking higher-order neighbours into account might also lead to search for unlikely very distant matches, because the Delaunay algorithm is range independent.

7.3.3.3 Radius based selection

We here compare all particles in the second view that are within a fixed radius of the particle in the reference view. This is a significant improvement for situations where disparities get higher compared to the distance between particles, like a global shift affecting the second view. In that case, even if the neighbourhoods are likely the same, they are located too far from each other to find them through the neighbourhood search.

This approach is more time consuming than the Delaunay based selection, because we have to set a high radius to be sure to match highly divergent points in sparse zone. Moreover, the matching is critical in sparse zones because the quantity of information is small and we cannot afford mismatches. This causes the number of selected points to increase in dense regions, thereby limiting the efficiency of the process. Note that the speed of the selection is higher because a distance check is shorter than a neighbourhood operation. However, it is not sufficient to compensate the increase in selected points. Still, we used this algorithm for its good results.

7.3.3.4 Distance comparison

Once plausible matches have been located, we need to compare them with a proper criteria. They are all based on the comparison of *stars* , i.e. the Delaunay represen-

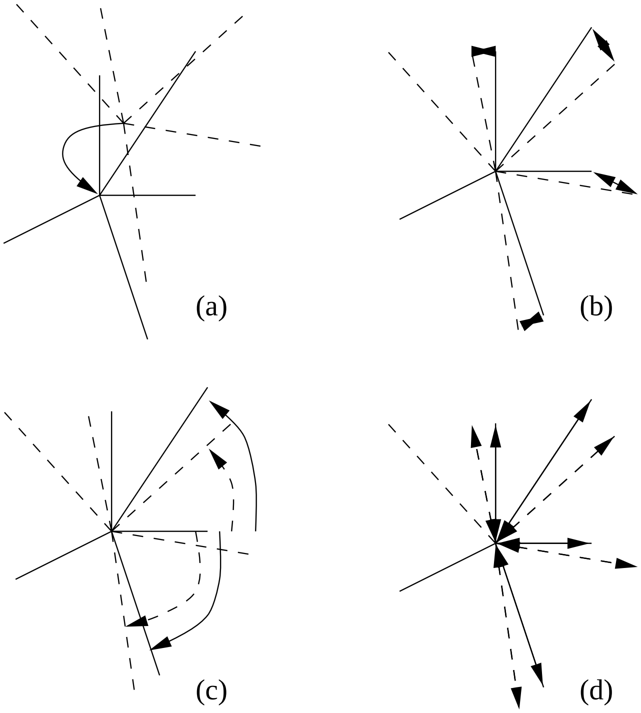


Figure 7.5: The comparison of stars: (a) change to relative coordinates, (b) distance match, (c) angles and (d) length comparison.

tation of the neighbourhood of a point.

Since we do not want the distance between the selected points to interfere, we work in relative coordinates: each point in the second image is moved, along with its star, to the point of the reference image we consider for matching (fig. 7.5a). The matching of stars is done neighbour by neighbour with an appropriate criteria.

In this first case, matching the neighbours is based on their distance to each other (fig. 7.5b): we compute the distance of each reference neighbour to each neighbour of the star from the second image, then match them using a minimal distance criteria with the algorithm presented at the beginning of this chapter for the search of best matches within an affinity matrix.

The next step is to compute the mean of the distances for the best matches, which is an inverted image of the similarity between the stars. In this process, we do not take the worse match into account because it usually comes from parasitic points in the mesh. This mean is used as the criteria for matching.

7.3.3.5 Vectorial comparison

We here take two informations into account : the length of the neighbourhood line (fig 7.5d) and its angle to the horizontal (fig 7.5c). The results are somewhat better, because we use all the 'phase' and 'amplitude' information available, while with the distance approach only the norm was computed.

The lengths and angles matrices are computed separately, then we multiply them term by term, and look for the best matches in the resulting disparity matrix A sum of the lowest disparities is done, just as in the previous method, to yield the matching criteria. It is the method we used in our analysis.

7.3.4 Conclusion of the neighbourhood algorithm

The first problems with this algorithm occur when parasitic points exists. We define a parasitic point as a point, particle or not, that is present on one image and not the other. Most of the background points detected as particles are parasites, but some dim particles, detected in one view and not on the other are parasites too. Such a point can considerably deteriorate the similarity between the stars, so we choose to eliminated them as we could. Therefore, all points that have no neighbours within a certain radius in the other projected view are considered as 'lonely', and are excluded from further search.

As far, this method has not yielded significant quality improvement compared to the exhaustive computation. It is faster, however, but one should keep in mind that it depends on the projection plane altitude. This process is well suited for very similar particle fields; it has thus been chosen to match particles between the two successive images of one camera, for position interpolation purposes (see Chapter 9).

7.4 The selected algorithm

We choose an algorithm that is close to the second solution, i.e. the exhaustive matching. The speed inconvenient has been avoided by checking particles within a limited radius. This introduces a slight altitude artifact, since particles with disparities out of the analysis range will not be matched. However, it is also an insurance against very bad matches (like two particles aligned with the epipolar line). A typical radius of 30mm has been used. At this point of the stereometric process, we have a list of tree-dimensional points supposed to belong to the fluid surface. We call this list a *forest*.

7.4.1 Forest composition

Because we suppose that the fluid surface is quasi-steady, we may concatenate all the forests from the successive image pairs. It is important concatenate the forests and not the position of particles in image planes, because the density of particles for matching would be much too large for an efficient matching. The number of three-dimensional points available for each pair of frame being around 300 and since we have 50 frames per run, the size of final forest before the surface regularization procedure is around 15.000. It is not necessary to use the quasi-steady surface hypothesis (one could work with one frame pair to recover a surface), but the gain in surface quality is very important.

7.4.2 Forest truncation

A simple improvement here is to eliminate all matches that are of a bad quality. It is equivalent to eliminate the worst of the best matches. This is done by leaving behind all matches with a distance between rays superior to a certain limit, which we choose to be 5mm. The spread function of the distance between rays for a complete concatenated forest is shown on fig. 7.6, while its density is shown on fig. 7.7. The average distance of eliminated points is 11.453mm, which shows that the abandoned fraction is of a bad quality and can be eliminated. Looking at the density graph, one can also see that it becomes erratic after 5mm, probably due to mismatches.

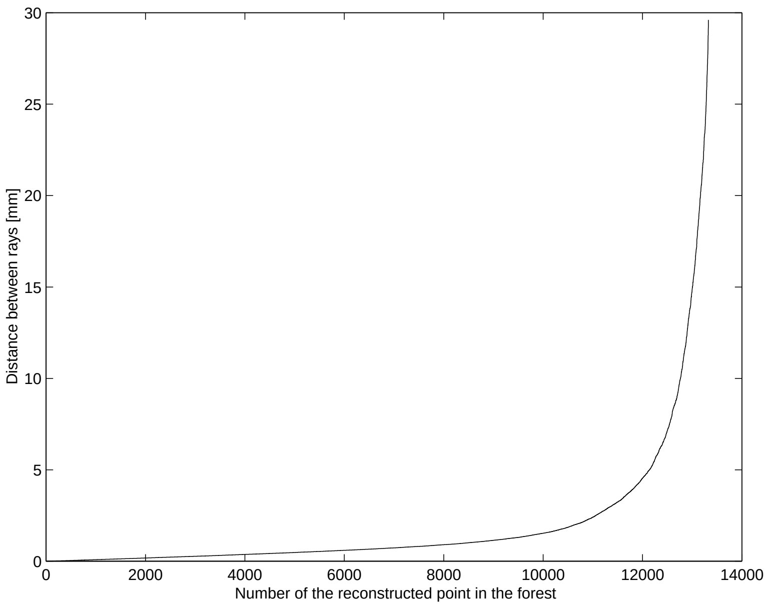


Figure 7.6: Spread function of the distance between rays of the final forest, run 3.

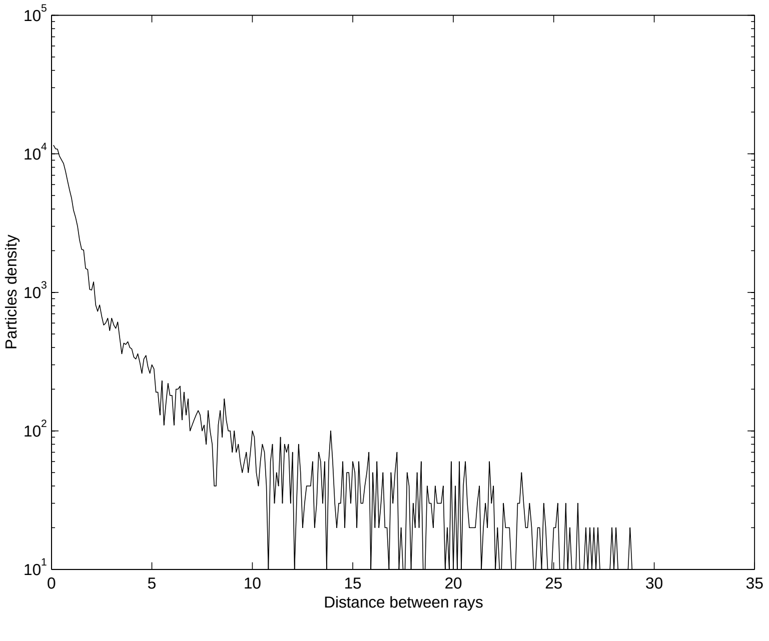


Figure 7.7: Density of particles as function of the distance between rays of the final forest, run 3.

Chapter 8

Surface reconstruction

Now that we have a list of 3D points, we want to reconstruct the flow surface. Once again, several methods were tested.

8.1 Interpolation

To be able to use the forest, we will place the points on a grid of limited size. This is done by cubic and nearest neighbour interpolation of the points of the forest near the grid point. Since we will use a filter, it is not necessary to use a more complex cubic interpolation. After setting the data to a grid, we will filter the surface. Because of possible mismatches and the scattered data, the surface is very peaky, and we need to apply a low pass filtering.

8.2 Simple filtering

A first simple method is to use a low-pass filter. In our experiments, we convolve the peaky interpolated surface with a gaussian window. Different window sizes for the filter were tested, with the best results with a 100mm×100mm size. This might look as an excessive size, but because of the gaussian shape only the points located in a 50mm×50mm tile will have a significant weight. The variance was set to 1/8 of the length of the filter, for the extremities of the filter are then nearly zero. We have thus only one parameter: the filter size. All filters are normalized so that their integral is one, yielding no surface height distorsion. Because the convolution involves less points on the borders and corners, we also have to multiply the results with a weighting map to restore the proper scaling on the boundaries (fig 8.1).

Another equivalent method based on FFT was developed, but no improvement could be achieved in execution time.

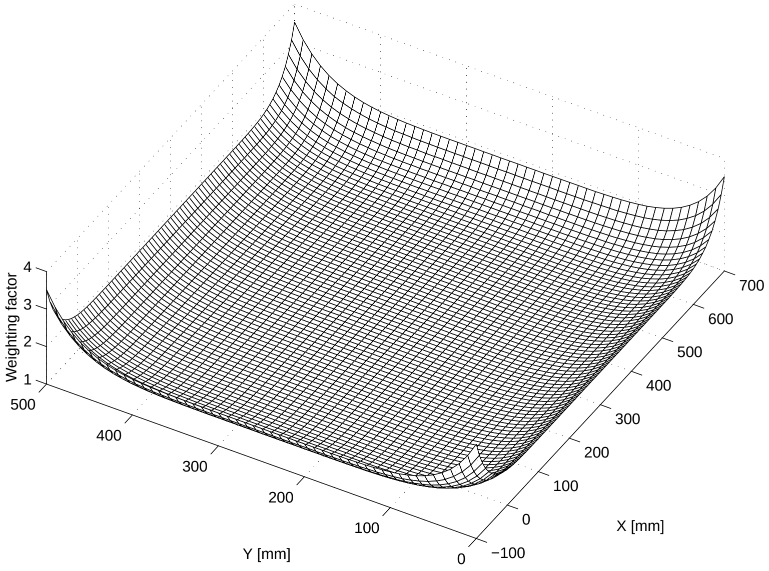


Figure 8.1: Compensation matrix for the boundaries.

8.3 Derivative method

Since the physical surface of the flow is expected to be sensibly smooth, we can reconstruct the thee-dimensional surface by minimizing the functional:

$$F\left(Z\left(X,Y\right)\right)=\sum_{k}\left(Z\left(X_k,Y_k\right)-Z_k\right)^2+\beta^2\int\int_{\mathcal{D}}\left(\nabla^2 Z\right)^2 dXdY \qquad (8.1)$$

where Z is the regularized surface elevation perpendicular to the mean plane, (X,Y) are the coordinates in that plane, (X_k,Y_k,Z_k) are the points obtained from the matching procedure, and β is a weighting coefficient that regulates the amount of smoothness desired. This variational problem can be cast into its Euler-Poisson differential formulation, which is then easily be solved by finite differences.

The problem of this approach is to define the necessary boundary conditions. Since the stereo process does not rely on any of these conditions, we did not use this method. If no boundary conditions are imposed, the surface may yield very interesting results in the center of the analysed domain, but the domain boundaries are then very unstable. It has been used at first for the velocimetric method, for which boundary conditions can be defined, but also abandonned, as explained in [7].

Chapter 9

The synchronicity problem

9.1 Introduction

Until now, we have supposed the exact synchronicity between the two cameras. Since the depth is determined from the feature points disparities between left and right views, it is clear that a synchronicity error will cause the particles to have moved during the lapse between the two takes, causing a dramatic decrease of the accuracy of the results. This problem is capital: an average shift as high as 40mm occurs in the 40ms interval between takes. Because the delay is at worst the half of the time lapse between takes, the maximal shift due to delay is $40/2 = 20$mm, which is the same order of magnitude as the average disparity between particles when matching. Not taking this problem into account would thus be a complete nonsense. It is the uttermost limitation of our results. In fact, and some of additions like subpixel accuracy when detecting particles are useless when comparing these to the lack of precision due to the synchronicity problem. We will see, however, that it is possible to reduce the effects of this problem and that the remaining inaccuracies do not affect significantly the final results.

The case of non-synchronous cameras is not common and is of course not recommanded. Because this work is intended to be a preliminary investigation for the validation of both the velocimetric and stereoscopic methods, we have integrated these additions to have a complete tool for stereometry, which should be used with synchronous cameras. The methods implemented to deal with the synchronicity problem can still be used then, with certain limitations, if a minor delay is observed.

Our goal is to get the positions of all particles (left and right) at the same instant, which can be done by interpolating the positions of particles between two consecutive images. Since the distances made by particles during the 40ms gap are very different, we cannot afford a simple X-axis translation, we will need to treat particles one by one and so track their movement to interpolate their positions.

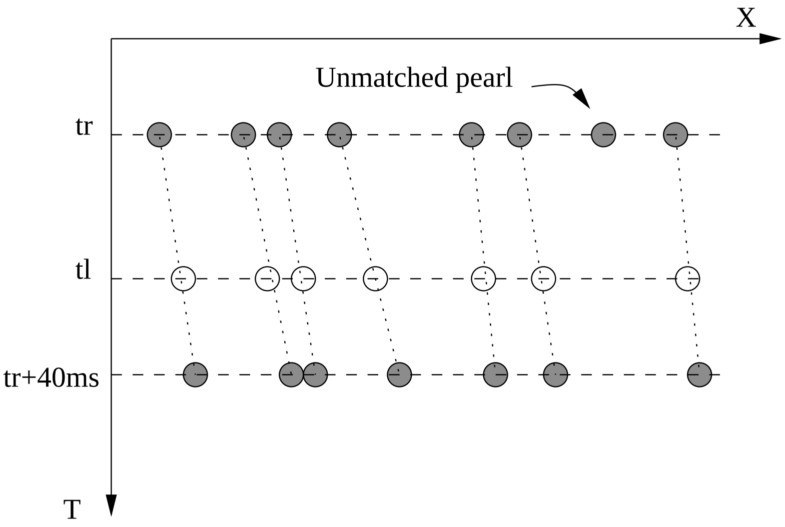


Figure 9.1: An unidimensional representation of the delay compensation. t_r and t_l are the times at which frames are taken by respectively the right and left camera, dotted lines indicate the linear interpolation of the particles positions and dashed lines represent the image planes.

We first present an external and internal method for delay estimation. We refer to the external term for the method based on data external to the flow, while we keep the internal appellation for intrinsic measurements of the flow. We will then show how to compensate the delay to recover synchronized particles positions. Note already that we do not need the absolute time where images were taken, but only the delay between these. Consequently, we can take one of the two series of images, say the left one, to be a reference, while the right one will pass through the synchronizing process to provide the particle positions at the same time as the left camera.

9.2 External delay estimation

The external delay estimation procedure is the following. Since no hardware signal (either input or output) was available from the cameras, we needed an external time reference to know when images were taken. Our idea is to put a counter in the field of view of both cameras: each frame will then bear the time at which it has been taken. This approach is elegant, but suffers some fundamental lacks.

First, the counter must be selected with care: the display may not be multiplexed since it would cause only some digit or even part of digits to be visible on each frame. We therefore designed and built a custom counter with no digit multiplexing. A custom counter also has the advantage to be of the desired shape so it can fit in an unused part of the scene without obstruction of the flow. The custom counter was available for the second set of experiments, and was placed on the flume boundaries. In the first runs, a commercial counter (still unmultiplexed) was placed over the flow, thus blocking a part of the scene that was lost. The different configurations can be seen on other figures throughout this text.

The use of simple counters has another problem. Because of the shutter speed (around 1/250e sec), the 1/1000e sec digit is unreadable: it is the image of the superposition of four consecutive values. We were able to determine the value of this last digit using the fact that segments that appear on more digital numbers will seems brighter. It is thus possible to correlate the brightness of the last digit with the theoretical brightness of a succession of four digits. A further improvement would be to change the last display into a 10 segment bargraph, the lapse would then be graphically visible, without the overlap of segments since they appear one by one, on different LEDs.

Table 9.1 shows the delays for the different analyzed sequences computed with this method. One can see that applying this method yields two perfectly synchronized runs (delay=0), which is very unlikely to happen. It is thus obvious that improvements must be find the delay with more accuracy.

	Run 3	Run 4	Run 5	Run 6
Delay [ms]	24	0	9	0

Table 9.1: Delays for the analyzed runs with the external estimation.

9.3 Internal delay estimation

9.3.1 A first attempt

It appeared when optimizing the interframe matching algorithm for delay compensation that it was possible to define an optimal shift between the views for the Voronoï algorithm. Because of the good results and properties for the estimation of that shift, we thought about applying the same method for optimizing the delays.

We have supposed that the use of the correct delay will cause the matches between rays to be more precise since the problem then approaches the ideal synchronized case. From there, we choose the average distance between the matched rays to be an image of consistency of the problem: the lower the average distance, the better the consistence, hence the better the estimated delay. Computing the variation of this distance for different delays yielded a delay of 33ms for the fifth experiment, far more than the 9ms computed above. Such difference is certainly not tolerable, so we looked for an explanation.

At this point of the research, a correct surface (in a qualitative way) for the third run was available and computed with the initial 24ms delay. The obvious way to verify the internal delay estimation was to check that for this correct third run the delays are the same for both the internal and external methods. The 24ms delay was found to be identical for both methods, comforting us in the use of the internal estimation of the delay. However, because of the great difference between the delay values for run 5, we had to find more precise criteria.

9.3.2 Revealing criteria

With these encouraging results, we looked further for other properties that might change with respect to the delay. We also added two special considerations.

First, to avoid noisy results, we worked with the concatenation of five successive forests. Next, in addition to working with the full concatenated forest, we also have taken into account the forest limited to its best matches. The selection criteria for selecting these best matches is the distance between the pairs, which had to be less than 5mm (this maximal distance allowance proved to be a good compromise). The fraction of selected points shows a peak at the location of the optimal delay (fig. 9.2). It comforts our hypothesis that the matches will be more numerous and more precise when the problem is consistent, i.e. when the delay is correct. In the next discussion, the properties computed with both the full and the restricted forest will be shown. The analysis of the different criteria will be done on run 3, we will then discuss the other runs in consequence.

9.3.2.1 Average distance

The variation average distance between rays of matched pairs, which already showed its potential, is represented on fig. 9.3. We see a clear minimum near 16ms, revealing the presence of the optimal delay.

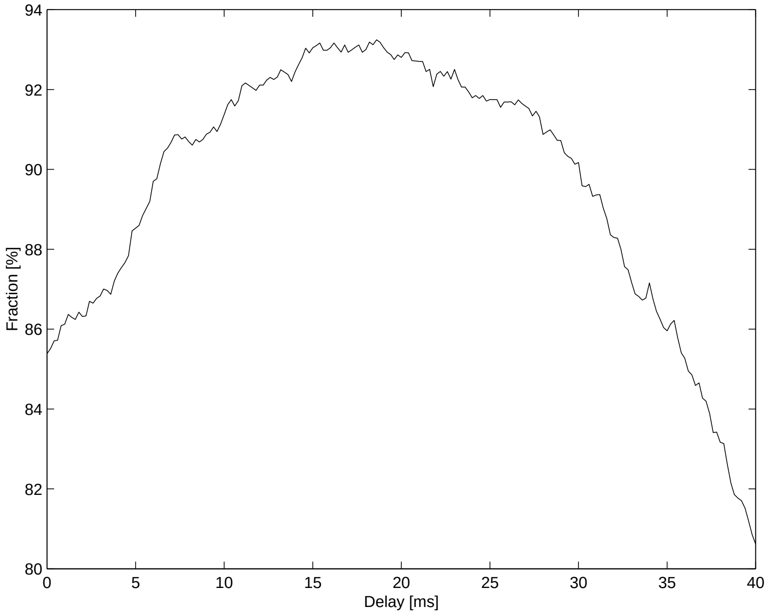


Figure 9.2: Fraction of matches for which distance between rays is less than 5mm.

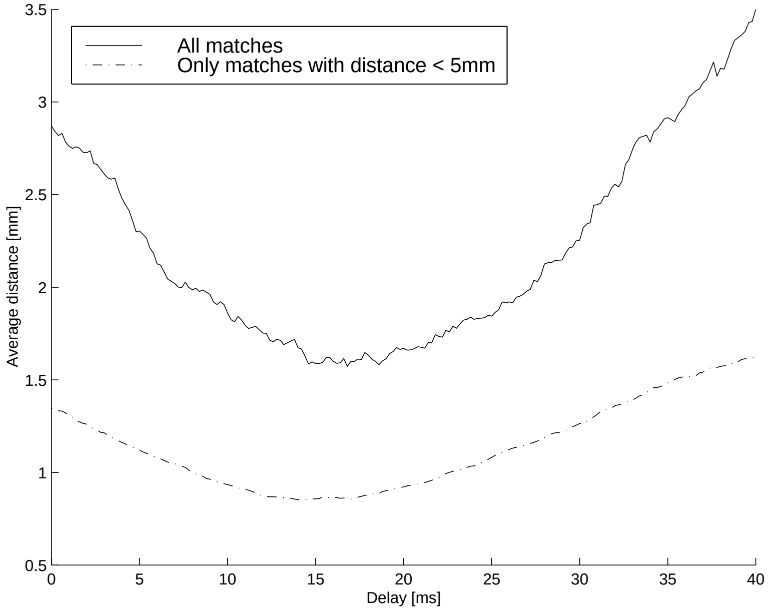


Figure 9.3: Average distance between rays of matched pairs for run 3.

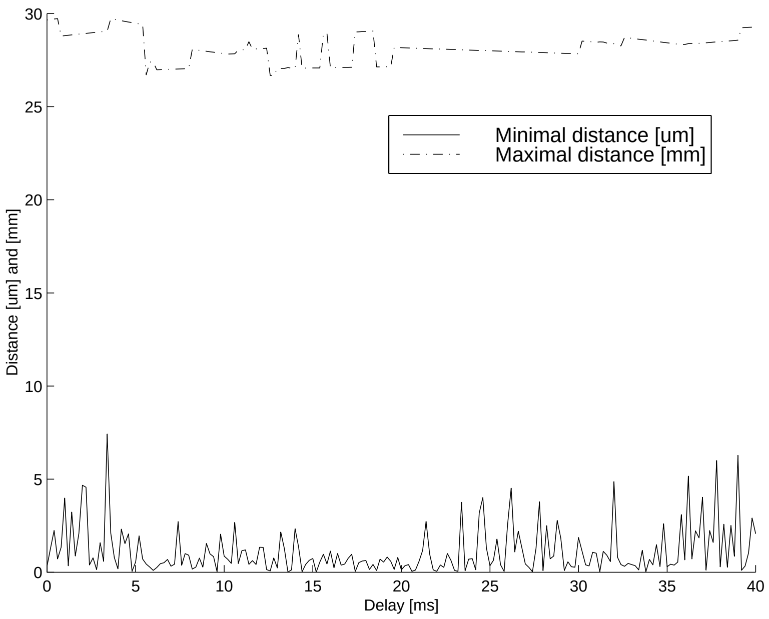


Figure 9.4: Minimum and maximum values of the reconstructed surface (run 3).

9.3.2.2 Minimal and maximal distance

Their variation is shown on fig. 9.4. The contrast with the average distance criteria speaks for itself: these properties are erratic and do not show any clear variation correlated to the delay. This is because they are based on one value (maximum or minimum) and not on an average, like the previous criteria. Given the bad behavior of these criteria, they have not been used further.

9.3.2.3 Number of matched pairs

This is historically the first criteria we used, for the best projection plane estimation and the average shift between two successive images. It shows the presence of a peak (fig. 9.5), like in the previous tests, and was thus also selected as a revealing criteria.

9.3.2.4 Amplitude of the reconstructed surface

The amplitude of the surface is computed as its maximum minus its minimum. Even if it is based on the reconstructed surface, it is just as peaky as the minimum/maximum distance, for the same reasons (fig. 9.6). It has been abandoned.

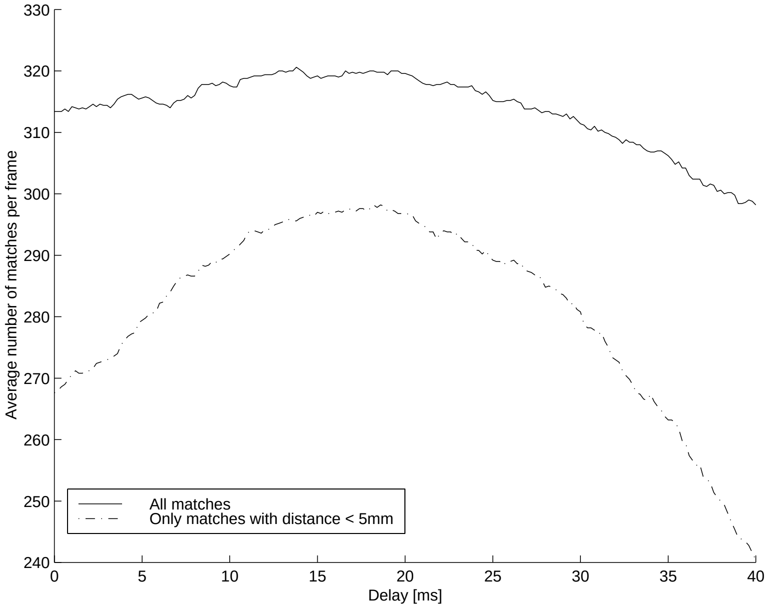


Figure 9.5: Number of matched pairs (run3).

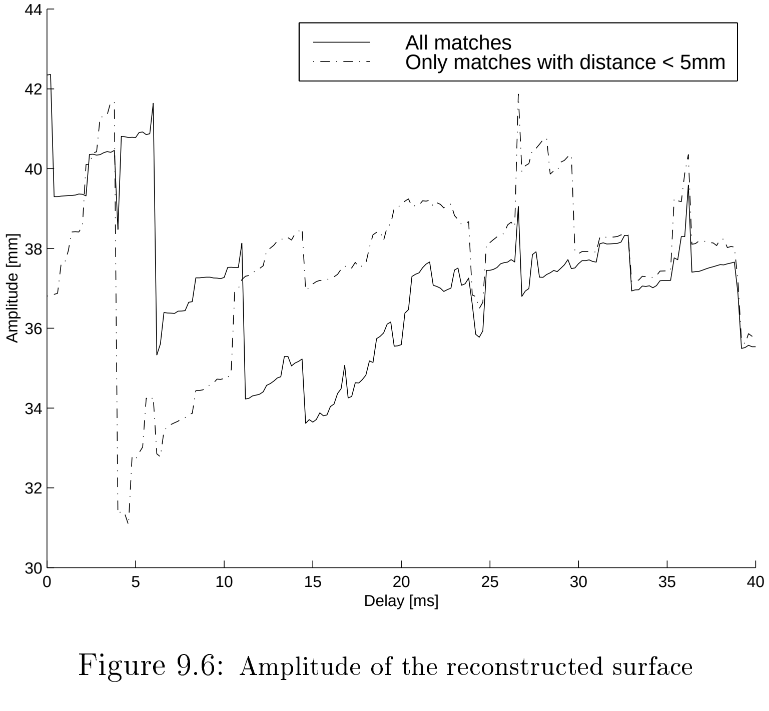


Figure 9.6: Amplitude of the reconstructed surface

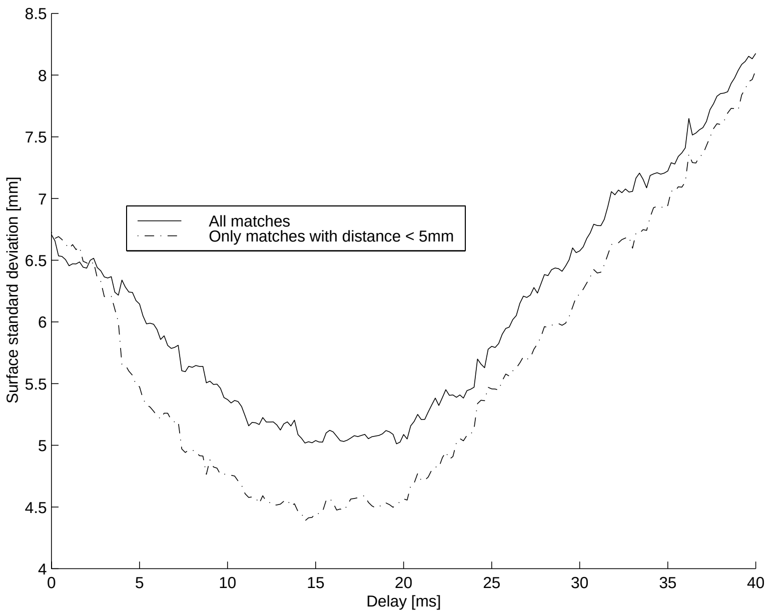


Figure 9.7: Standard deviation of the surface compared to the forest.

9.3.2.5 Standard deviation

We here measure the quality of the surface. This is done by computing the average distance between the reconstructed surface and each point of the forest. For an optimal delay, the consistence of the problem will lead to less noisy results and very distant points due to mismatches. We thus expect the plot to show a minimum for the optimal delay, which is the case, as shown on fig. 9.7. This criteria was thus also chosen as revealing.

9.3.3 Combination of criteria

We now have four valid criteria: the average distance between pairs, the standard deviation of the surface, the fraction of best matches and the number of matched pairs. Since the curves obtained with the restricted forest seems smoother, we used them instead of full forest curves.

Recombining the criteria requires a scaling, since properties vary in very different ranges of values. We therefore subtract the minimal value of each property, then divide the result by its maximum. This yields a series of curves with values ranging from zero to one. We also invert the two properties that show maxima before adding them, so that all criteria are minimal at the point of the optimal delay (note, however, that we will always show these properties in their original form). The curves are the

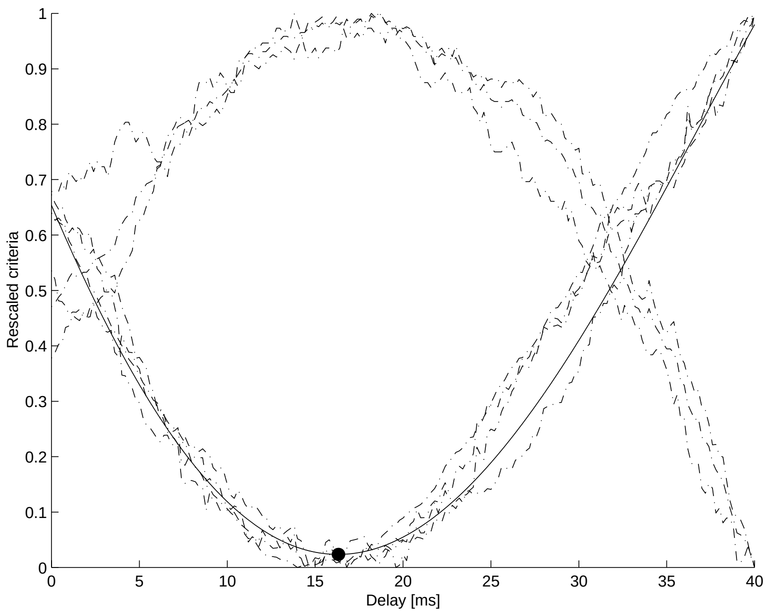


Figure 9.8: Interpolation of the sum of rescaled properties, run 3.

added and a polynomial fit is conducted. The best order for the interpolating polynomial was found to be four. The interpolation, together with the added properties curve, is shown on fig. 9.8. The optimal delay for the third run is 16.31ms, far from the first estimation of 24ms, but much more precisely computed. The final resulting process is shown on fig. 9.9.

9.3.4 The other runs

9.3.4.1 Fourth run

The final graph (fig. 9.10) is very different from the third run: all properties vary continuously in the same way. This may mean that we are not working with the correct frames, i.e. that the best delay is negative. We recomputed the graphs using frame -1 and frame 0, this is shown on fig. 9.11.

The result is unexpected: though the variation is in the other way, which is logical, there is still no global peak in the data showing an optimal delay. The only conclusion is that the delay is zero with our precision. We could have tried to refine the results to find the exact (small) delay, but it is pointless for two reasons.

First, the delay we will find is going to be very small, the correction being probably very negligible. Secondly, the process of synchronization itself is nor error nor cost free: there exists mismatches with the Voronoï algorithm (mismatches will lead to a

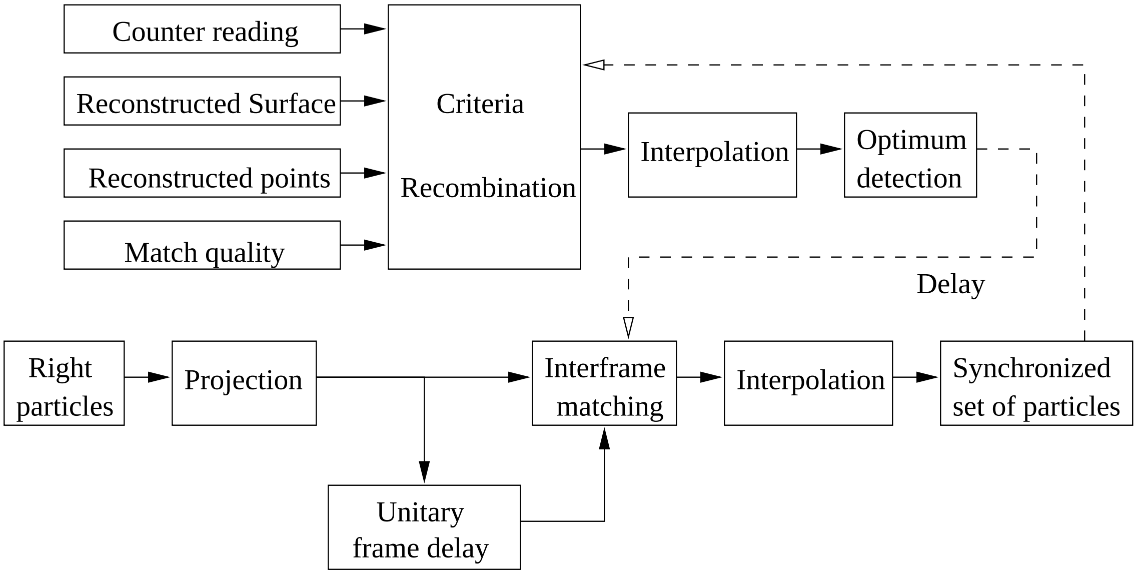


Figure 9.9: The internal synchronization process.

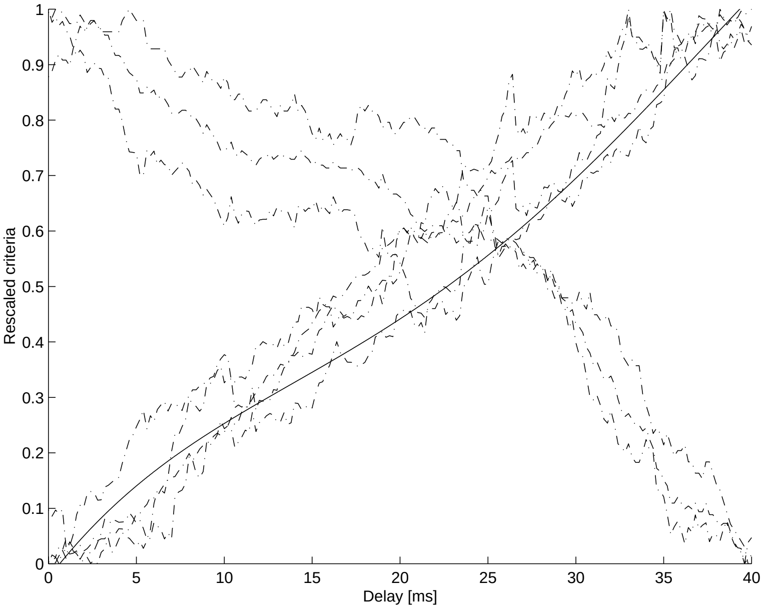


Figure 9.10: Interpolation of the sum of rescaled properties, run 4.

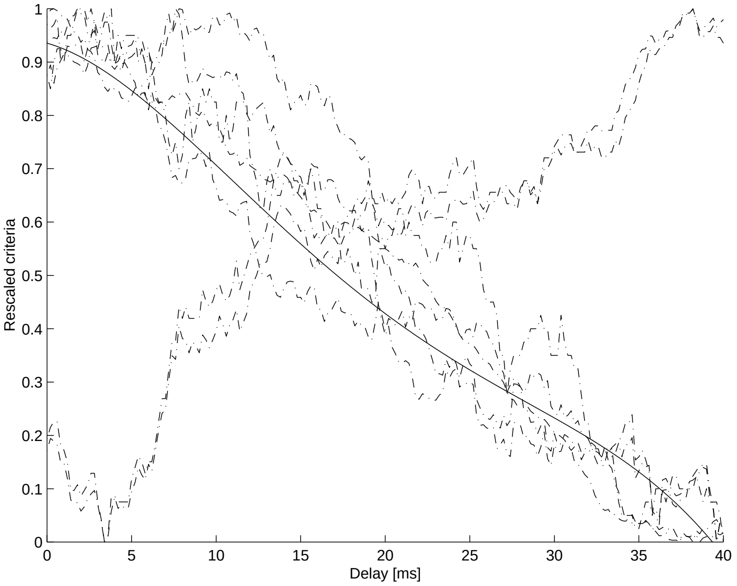


Figure 9.11: Interpolation of the sum of rescaled properties with the previous frame, run 4.

wrong particle movement within the delay and this effect might cancel the synchronization benefit), and also lots of particles remain unmatched, and are lost. Moreover, run 4 was incomplete, and we certainly do not want to loose particles for the very small (if ever) benefit of a synchronization process. We worked thus with the fourth run as if it was synchronous.

9.3.4.2 Fifth run

The graphs on fig. 9.12 lead to a delay of 24.87ms, less than the first 33ms estimated, but this time it has been computed more rigorously. An a posteriori check of the final surface for this delay confirms the internal evaluation process.

9.3.4.3 Sixth run

We here face the same problem as run 4: the minimum is located at one end of the interval (fig. 9.13). We thus follow the same process, and look for a optimal delay with the previous frame. This time, as shown by fig. 9.14, we locate a minima corresponding to an optimal delay of 31.72ms, which means, if the internal method is consistent, that the frame matching made with the external counter was wrong.

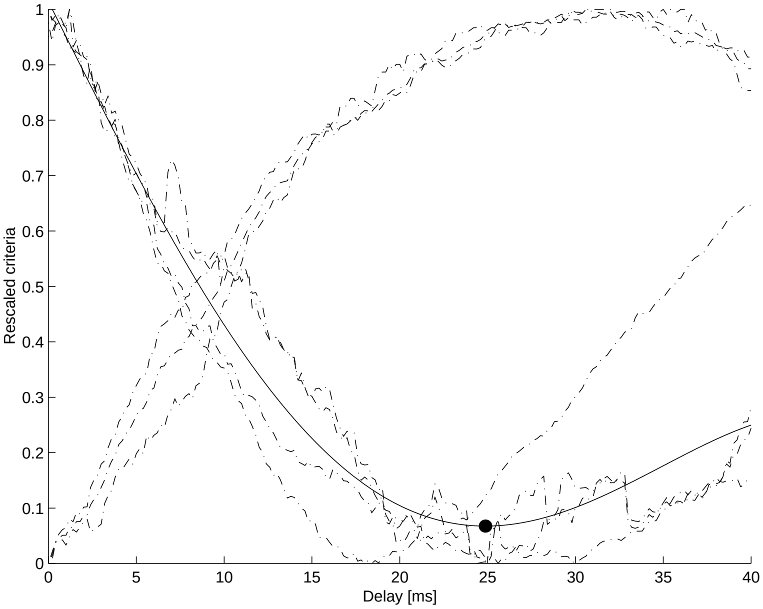


Figure 9.12: Interpolation of the sum of rescaled properties, run 5.

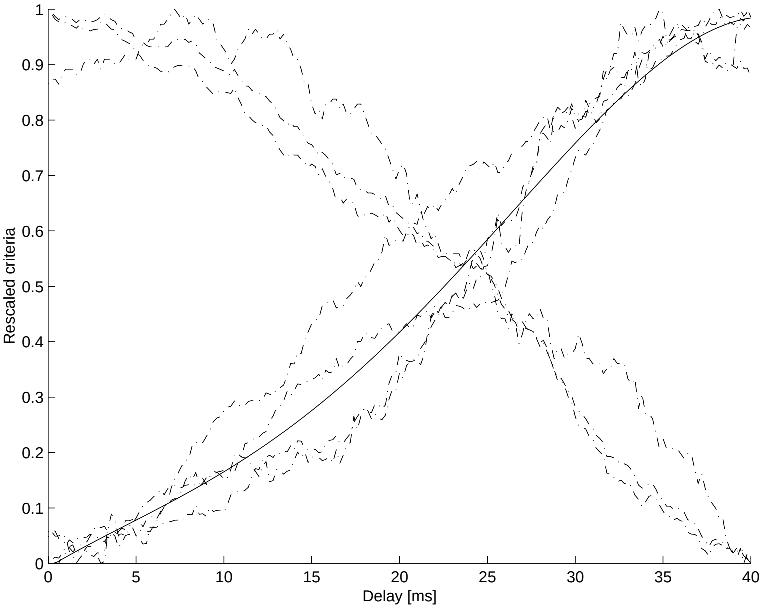


Figure 9.13: Interpolation of the sum of rescaled properties, run 6.

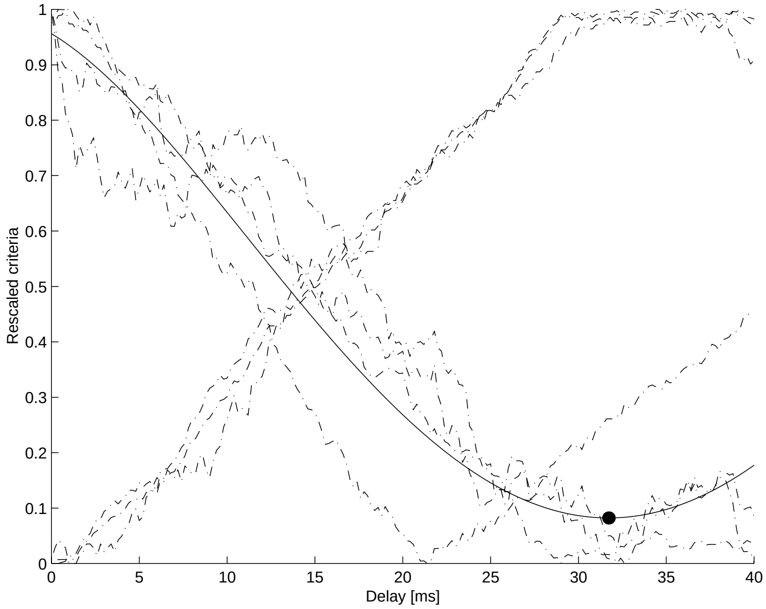


Figure 9.14: Interpolation of the sum of rescaled properties with the previous frame, run 6.

9.3.5 Conclusion of internal and external delay estimation

With the sixth run, the internal delay compensation has shown all its potential: not only have we been able to estimate the delay, but we also found that the first frame matching with the external counter was wrong, due to a reading error. Providing that the combination of criteria used here is revealing (which is confirmed by the relative quality of the final surfaces obtained), we can conclude that the machine has been able to correct a human visual inspection, which is very satisfying for an image processing problem...The final delays we will use in the next developments are presented below in table 9.2.

	Run 3	Run 4	Run 5	Run 6
Delay [ms]	16.31	0	24.87	31.72

Table 9.2: The final estimated delays with the internal delay estimation method.

9.4 Delay compensation

9.4.1 Interframe matching

As stated above, the delay compensation requires an adequate matching of the particles over the time. The similarity of the views, due to the fact that they are taken from the same camera, enables us to use the Voronoï algorithm to match successive views. This method is attractive because of its originality and speed. We will here focus on the context of its application.

9.4.1.1 Global shift estimation

The matching is based on neighbourhood operations, and we thus want to have two similar images. We will first apply a constant shift to one set of particles. The amplitude of it is set for maximum performance of the Voronoï algorithm. This shift can be seen as a passage from a DC to an AC signal: we remove the constant shift, and let the Vorono algorithm focus on the AC component.

The shift has been adapted by trying different possibilities, but it is possible to automate the optimization. One could try one shift, then compute pairs using Voronoï matching. The quality indicator is the number of matched pairs. By iterating the process, it should converge to an optimum. However, this method is rather heavy. We simply computed the number of matches for several shifts, and apply the best one to the whole sequence. The shift will not be different for the successive views because the flow rate is approximatively constant during the experiment. We can thus apply the same shift for the whole run.

Optimal shifts for the different runs have been computed using a dichotomy process and are shown on table 9.3. The results are also a view of the speed of the flow and its flow rate. Their consistence is confirmed by table 3.1.

	Run 3	Run 4	Run 5	Run 6
Mean plane altitude [mm]	41	36	54	42

Table 9.3: The mean shift necessary for Voronoï matching between to views of the right camera.

Applying a constant shift cannot be done with the original particle's image positions. Indeed, the image plane is not parallel to the flume bottom plane, while the global speed vector of the fluid is. Applying a constant shift to the image coordinates would thus be an aberration since its value reported to a plane parallel to the mean flow level would not be constant. We need first to project the particle's positions onto an horizontal (in world coordinates) plane. The height of the projection plane is not revealing, since we only work here with one camera (only the scale changes, it could be seen as changing the focal distance). It has been fixed to 25mm, approximatively the mid-level of the fluid surface.

9.4.1.2 Parasites elimination

Before applying the Voronoï algotrithm, we want to eliminate all points that are not likely to be matched, i.e. that have no close neighbours in the other picture. The word 'close' can be replaced with an experimental distance of around 1.5 times the search radius. These synchronization points, corresponding to pearls detected on one image and not the other or to parasitic detections, create lots of distortions on the Voronoï polygons, decreasing the efficiency of the algorithm.

9.4.2 Interpolation

The application of the Voronoï algorithm leads directly to a list of pairs. The intermediate positions of particles we want will be obtained by interpolation. We use a simple linear interpolation over two frames. It is possible to use a higher order interpolation which requires to track particles over more than two frames, but the longer the track, the less the number of particles paths. This number eventually decreases dramatically, given the high noise level of the problem, mismatches,... We can note for future improvement that tracking particles is the core of the velocimetric method and its results could be imported here.

We now have a set of particles for the left and right camera, as if the cameras were synchronous, eliminating the effect of parasitic delays when matching and estimating the tree-dimensional coordinates of particles.

Chapter 10

Noise considerations

10.1 Synchronicity

We will discuss here the effects of a synchronization fault. Using asynchronous cameras has effects on both the correct matching of particles and the quality of the resulting depth. We will study the effects of both of these.

We can expect the resulting depth to change as the delay changes, since this would mean the displacement of one set of particles (left or right) with respect to the other. A qualitative approach was conducted, to see the effects on the computed surface.

We first compute the pairs of particles for a certain delay. These pairs are then fixed, and we change the delay (hence the interpolated particle position) to see its effects on the surface. It appeared that the surface did not changed significantly. If we now let the pairs change when varying the delay, the surface changes very much at certain locations. The problem of synchronicity appears then to be a matching problem rather than a change in the reconstructed points.

This strange effect can be explained by the selected matching algorithm: only the distance between pairs has been considered for matching. Even if a synchronization fault occurs, there will always be particles able to match with others: a matched particle moving away will be replaced by another one just aside it, because the particle density is high in nearly every region of the observed scene.

10.2 Detection noise

As stated above, we used a subpixel precision method to locate the particles. It is thus of great interest to see if the subpixel accuracy is achieved or not. We therefore computed the effects of random shifts applied to the positions (in the image planes) of the detected particles. For the further discussions, we will use an artificial horizontal

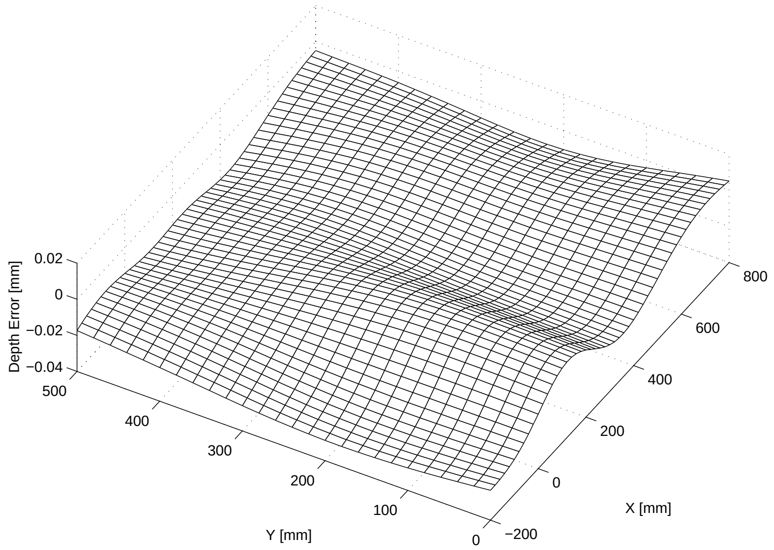


Figure 10.1: A random noise distorted plane for calibration 5.

plane at an altitude of 25 mm as reference surface. The right image coordinates of its points is slightly changed, representing the error. The whole stereo process is then conducted, except for the delay compensation, and we see to what extent the applied shift changes significantly the artificial plane. The resulting plane for a random white noise of one pixel in amplitude is shown on fig. 10.1. The error is very low, due to the smoothing of the surface regularization process. It shows, in the end, that the subpixel precision method developed above is not necessary to achieve good accuracy.

10.3 About the camera positions

Two camera positions have been used for the November runs. Positioning the cameras is a compromise between the X-axis and Z-axis precision. One can show that two cameras placed such as to face each other (closer to horizontal) will have the best resolution along the Z axis (vertical). On the other hand, cameras placed parallel to each other (closer to vertical) will have a good X location precision, but no Z precision.

The test used is the same as the random noise test described above, but the shifts are now of a fixed amplitude. Results are shown on fig. 10.2. We note that differences exist between an Y-axis shift and an X-axis shift, both in the shape of the resulting plane and the amplitude of the error, the latter being significantly lower for the Y-axis displacement. This is because a change along the Y-axis will not alter the depth

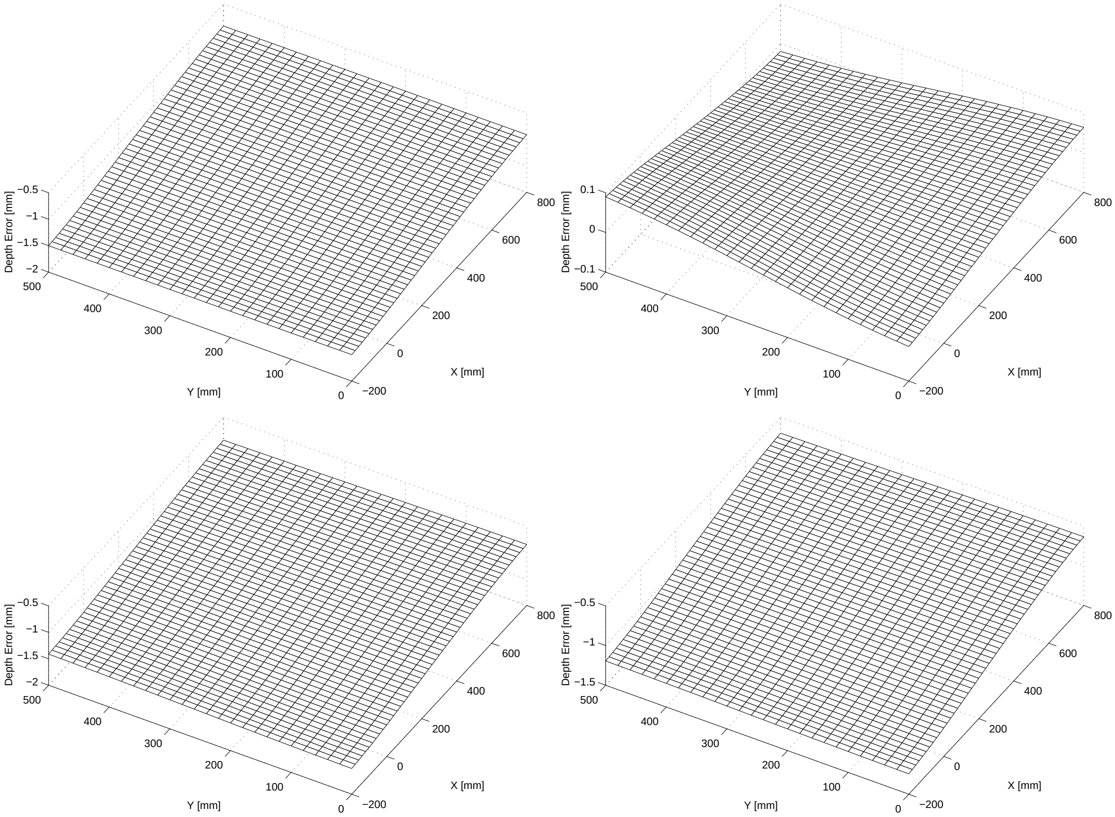


Figure 10.2: Different error planes: calibration 5, upper left: 1 pixel shift in the X direction, upper right: 1 pixel shift in the Y direction, lower left: combined shift in X and Y directions. Lower right: calibration 6 with the combined shift.

estimation but the Y-axis location. We also note that if the cameras get closer to the horizontal (calibration 6), the error on the resulting Z altitude is lower, which confirms the theory.

Part IV

Conclusions

Chapter 11

Final results

11.1 General comments

The process described in the previous sections led to surfaces for the analyzed runs 3 through 6. We first note that for all these, the stereometric and velocimetric methods yield very similar results, both in a qualitative (the surfaces outlines) and quantitative way (similar amplitudes, phase and wavelength). Because of the strict independence of the two methods, we can conclude to their efficiencies.

One can also observe that the falling slope of antidunes is steeper than its rising slope. It is confirmed by the velocimetric and stereoscopic methods, and also by large-scale natural events, like the interesting Burdekin River case presented in [26].

11.2 Third run

The surface was made of rolls, with their principal axis perpendicular to the flume axis. This is confirmed by the two methods.

We note that the rolls get smaller when going downstream. This is probably because the flow is very sensible to the blade present at the end of the flume. This blade, which keeps the sediment level to a certain height, must be horizontal. If not, a dissymmetry in the surfaces and bedforms will occur, which is the phenomena observed here. The presence of this variation, though we had precisely aligned the blade, illustrates the high sensitivity of the flow with boundaries and intruding objects. Another change in shape with respect to the position of the antidune is shown on picture A.6. The surface for run 3 might be compared with picture A.3.

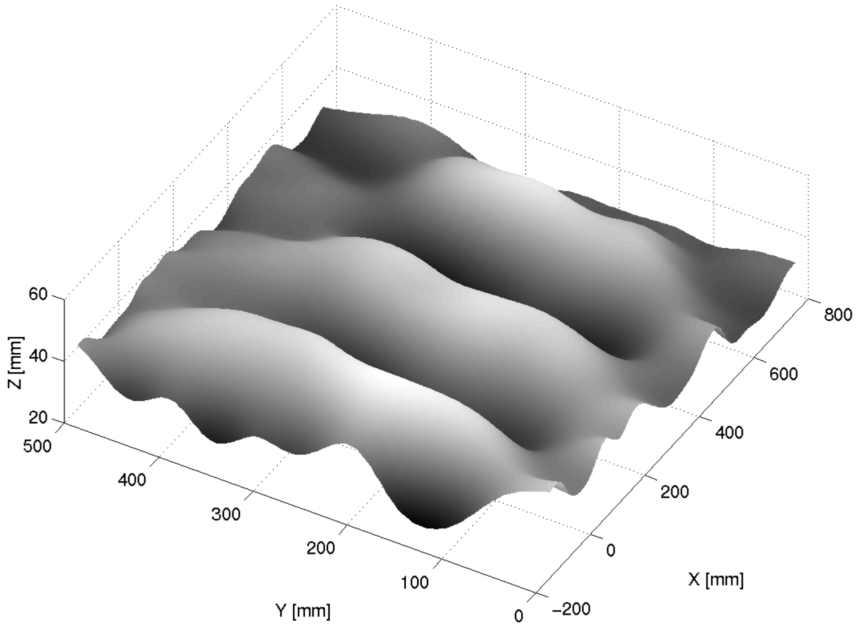


Figure 11.1: The final reconstructed surface for run 3.

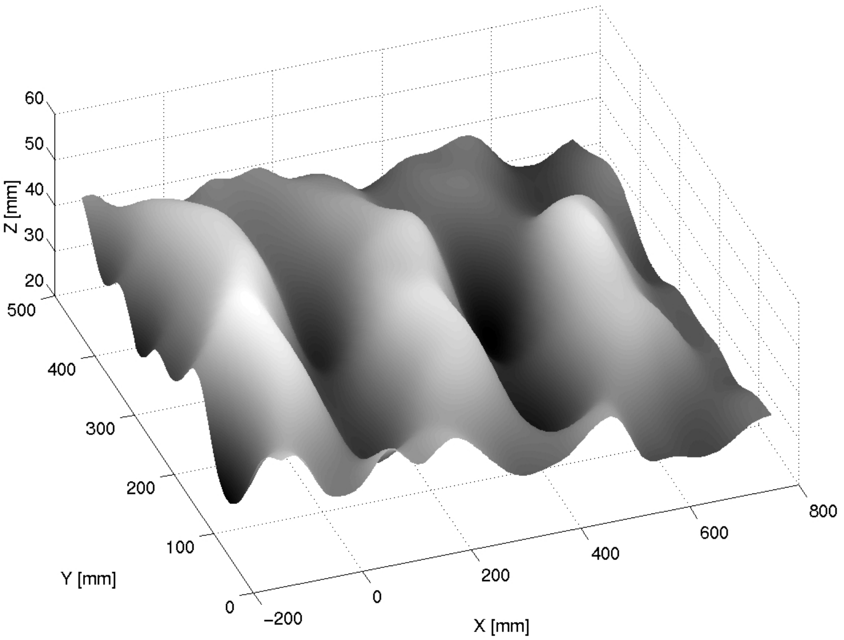


Figure 11.2: Another view of final surface for run 3.

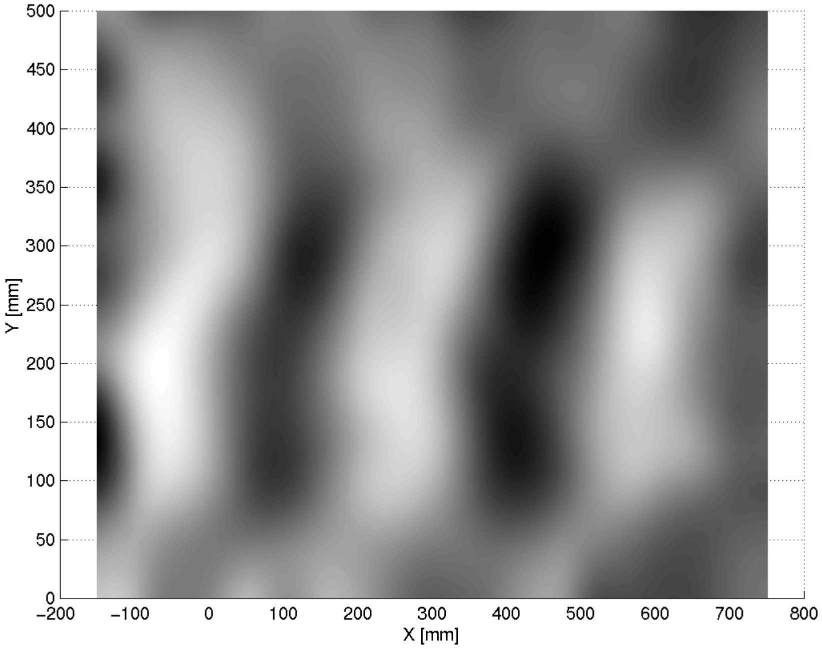


Figure 11.3: The top view of final surface for run 3.

11.3 Fourth run

Note that for the fourth run, the image sequence for stereoscopic analysis was incomplete, and only a part of the surface could be reconstructed (the part in which particles were present).

The analyzed pattern was a series of peaks placed regularly on the central axis of the flume, which is coherent with the resulting surface and can be compared with picture A.2.

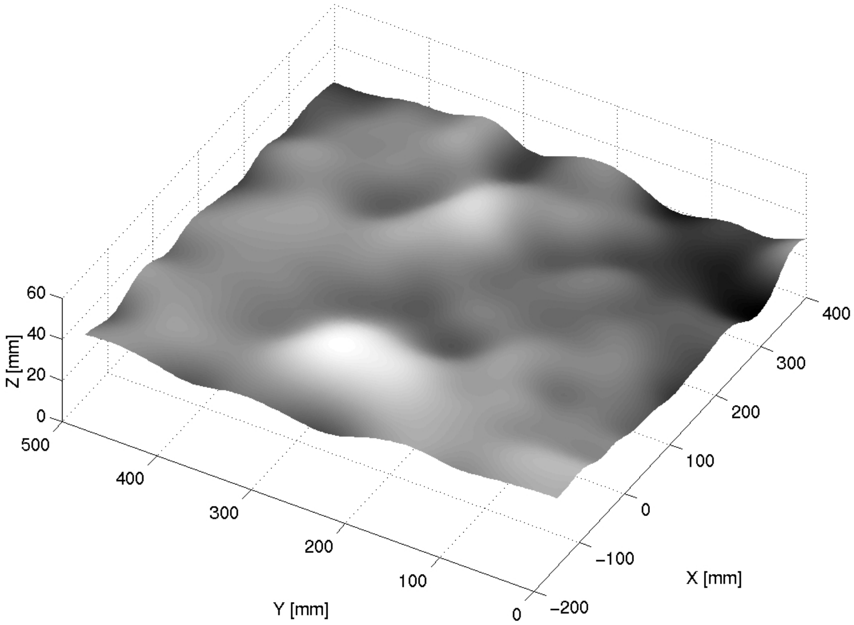


Figure 11.4: The final reconstructed surface for run 4.

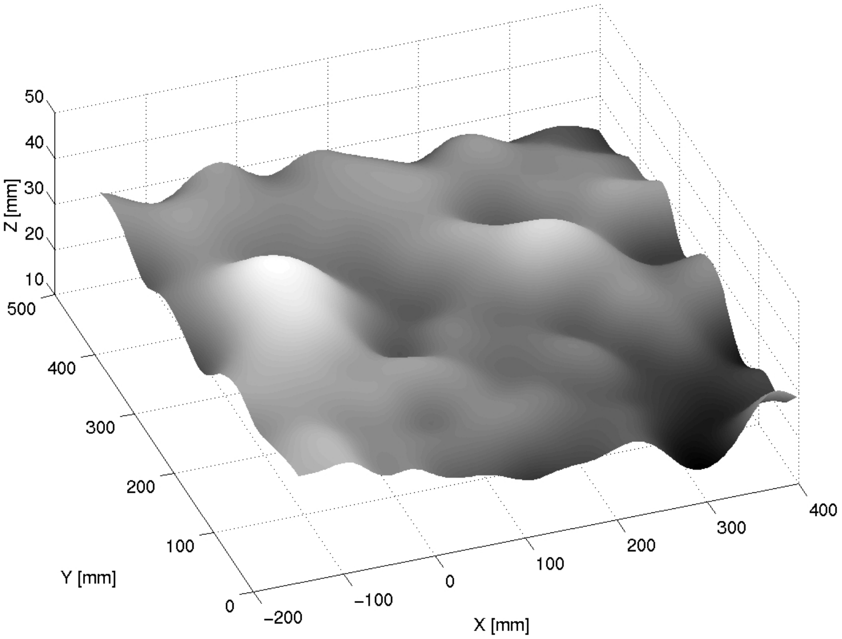


Figure 11.5: Another view of final surface for run 4.

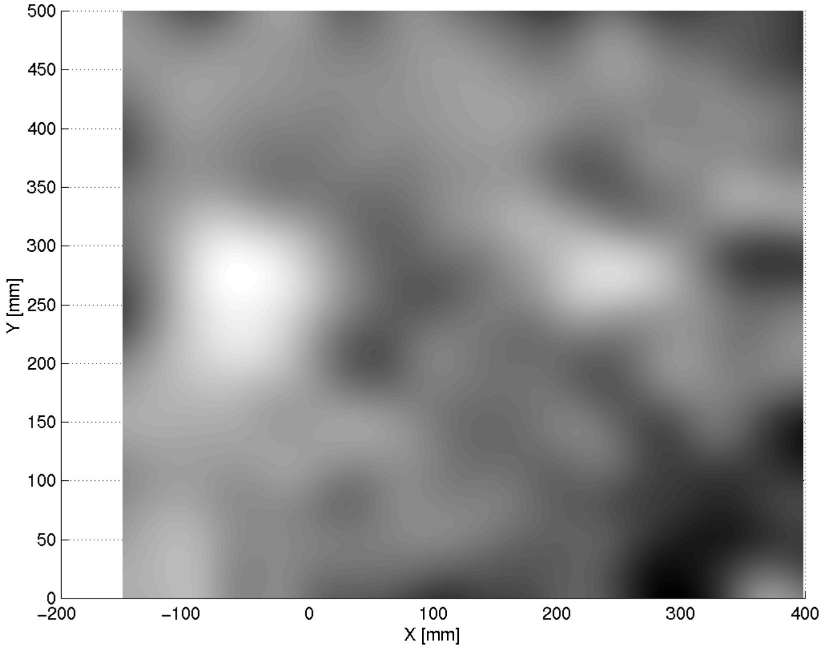


Figure 11.6: The top view of final surface for run 4.

11.4 Fifth run

This surface presents an intermediate result between surfaces 3 and 6. One can recognize the long patterns of rolls, but these are less straight, even more looking like vee's, which is the case of run 6. Once more, we note the dissymmetry in amplitude between the two sides of the flume which is associated to the same effect as run 3. An additional dissymmetry is that the higher-Y and lower-Y antidunes are in phase opposition. One can compare this surface with picture A.6.

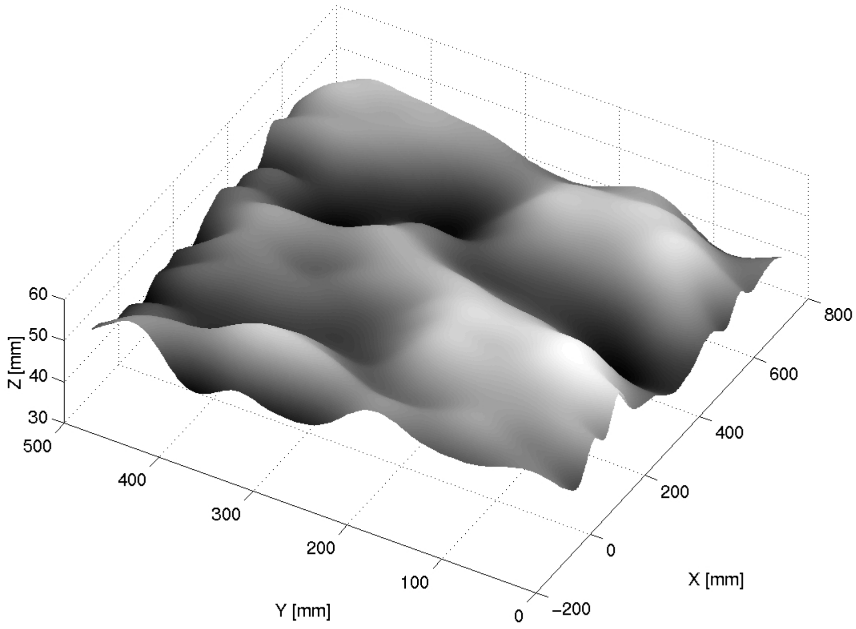


Figure 11.7: The final reconstructed surface for run 5.

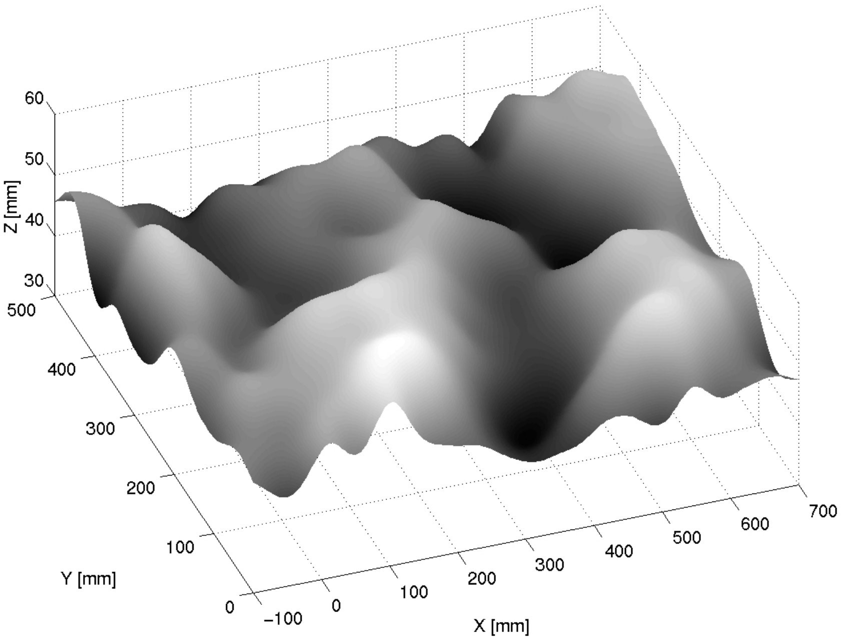


Figure 11.8: Another view of final surface for run 5.

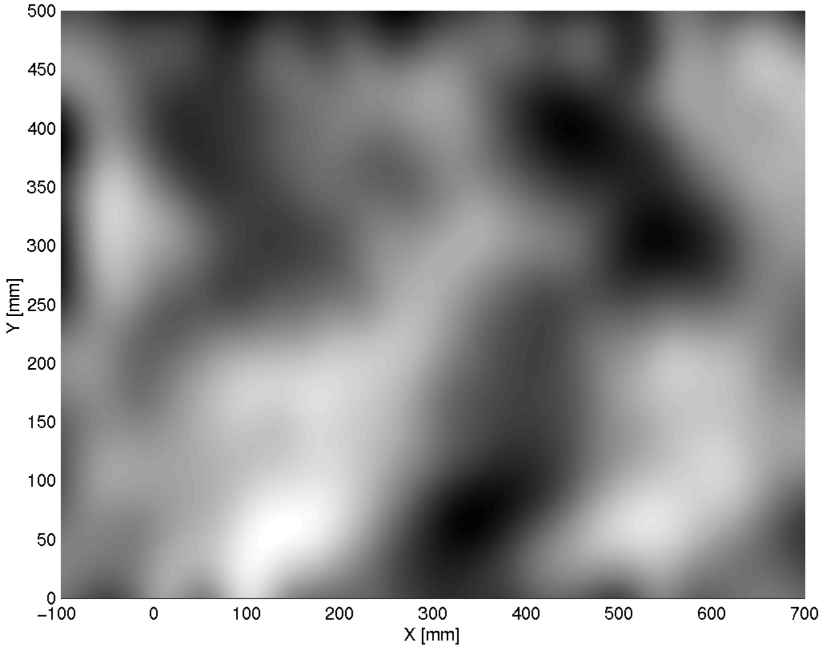


Figure 11.9: The top view of final surface for run 5.

11.5 Sixth run

The surface consisted in a complex two-dimensional pattern, which a human eye could but difficultly identify. The results show that it is made of the superposition of two vee's, their peak being located near one third and two thirds of the flume width. To be compared with photo A.4.

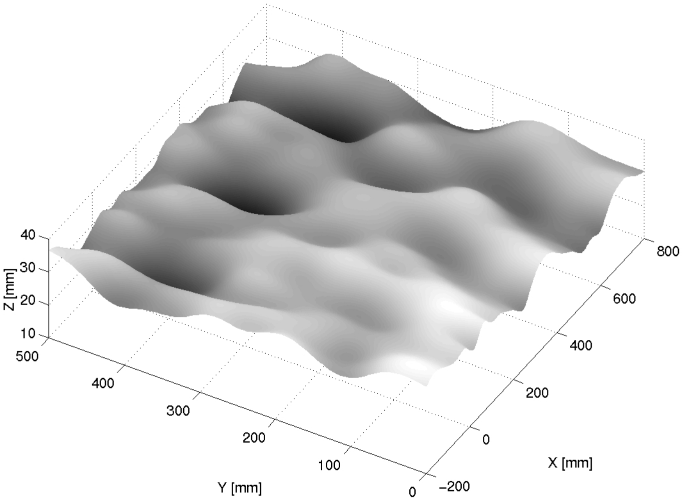


Figure 11.10: The final reconstructed surface for run 6.

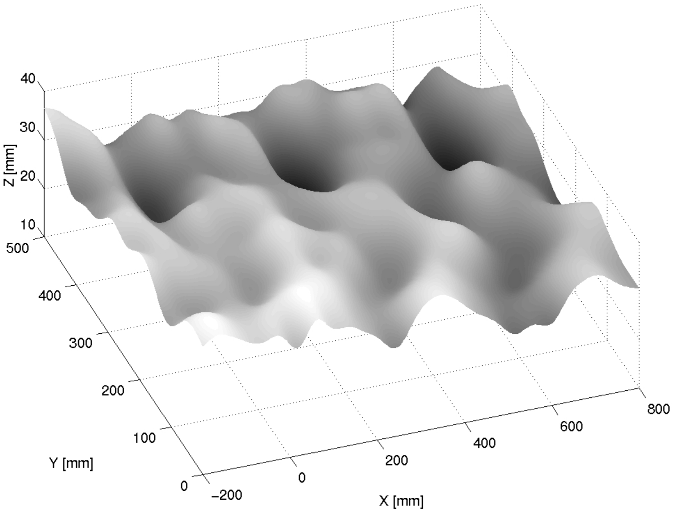


Figure 11.11: Another view of final surface for run 6.

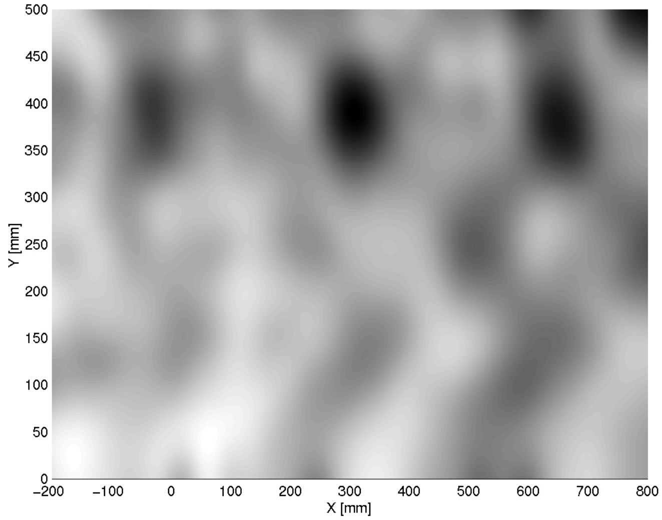


Figure 11.12: The top view of final surface for run 6.

Chapter 12

Future improvements

Numerous future improvements have been introduced throughout this text. We will here only focus on the most important ones.

12.1 Iterative processes

The processes used for our stereometric algorithm is not iterative, not even part of it. This is due to the already high computation time. However, several improvements can be made in that direction.

12.1.1 Voronoï matching

An iterative improvement for the Voronoï matching algorithm used for delay compensation would be to apply a variable shift, instead of the fixed shift envisaged. In that case, a first match with a fixed shift would produce a first approximation of the displacement vectors in a certain region. One could then apply a shift proportional to the displacement vectors in the neighbourhood of the considered point. Because displacements, or rather their AC component, are very different from region to region, it would most certainly yield a better match.

12.1.2 Matching and depth using the a posteriori computed surface

For the matching process, we could integrate another iterative improvement that would consist in taking the final surface into account. The affinity matrix would then integrate, for each match, the distance between the resulting three-dimensional point and the reconstructed surface. The iterations are launched with only one criteria (the distance between rays) and when the surface is available, we recompute the matches using the latter data.

12.2 Higher order interpolation for delay compensation

The interpolation process for delay compensation only used two frames, hence the interpolation was limited to the first order. Using more frames would allow us to compute a higher degree, more accurate interpolation. Such improvement would take benefit of the high-speed camera and the algorithms developed for velocimetry, since these involve tracking.

12.3 Multicamera system

As we already pointed out, the whole experiments could use the three cameras for depth estimation because the delays between takes from any camera is known (see section 4.3.3). In that case, several solutions can be adopted.

A first idea is to compute matches between each camera, the only keep those for which the direct match, say left-right, is identical to the transited match, left-upper-right. The idea is thus analogous walking on a triangle from one corner to another, using the two possible paths. It is also possible to incorporate the data from the third camera into the affinity matrix in an iterative process like the previous ones.

Another idea, much more time-consuming, would be to compute the affinity between the three set of points from the three cameras. The affinity matrix is the a three-dimensional matrix, and the match criteria must be changed. This would probably not give interesting results, for the upper camera is less precise than the left and right ones. Hence, the upper camera should be used only for matching, not for computing the three-dimensional coordinates.

From the point of view of mismatches, the upper camera could probably detect those due to an alignment with the baseline, as illustrated on fig. 7.2.

12.4 Including the Bernoulli equation

Since we know that the fluid surface follows the Bernoulli equation for its constitution, it would be interesting to take this equation into account when reconstructing the surface. We did not used it here because we want the two stereometric and velocimetric methods to be independent. Now that their efficiency has been shown, we could add such improvement.

Because the Bernoulli equation links the Z coordinate of the surface points to the local velocity, it should be possible to add an additional criteria in the affinity matrix used for matching. For instance, we can compute the local velocity with the interframe Voronoï matching, then apply the Bernoulli equation to obtain the probable Z coordinate (not accurate, for the velocity estimation is approximative with the stereo cameras. One could also add here the velocimetric measurements, see [7].). This idea of the Z coordinate of the point of the surface could be used as an additional matching criteria.

Chapter 13

Conclusions

We presented a stereoscopic method to analyze the complex surface created by a fast flow over three-dimensional antidunes. Such analysis was until now thought as out of reach by theoretic authors (see [27]), or made with intruding objects, perturbating the very sensible flow.

However, we showed that it was possible, and even in two different ways (see also [7]), to capture the water surface without significant artifacts. The stereometric and velocimetric methods, which are completely independent, yielded strongly similar surfaces, comforting us in the quality of our results.

One could say that the results presented here are approximative because of the lack of synchronicity between the cameras. While this is certainly the utmost inaccuracy of the process, we would rather point the robustness of the whole method that showed it was possible to cope with a simple synchronization algorithm. It should also been said that, while synchronicity can be easily achieved in a laboratory environment, the study of large-scale flows requires spaced sensors which are more difficult to synchronize (e.g. airborne cameras). It is thus of great interest to see that unsynchronized cameras can still provide useful informations when combined with our process.

From an hydrodynamic point of view, we could say that the study has also been a success, for four different surfaces have been reconstructed: rolls (run 3), peaks (run4), and two more complex patterns (run 5 and 6). Each of the observed surfaces corresponds to a specific state of the flow along the flow rates, the latter being varying or steady. The diversity of antidune patterns created by the flow could now be more thoroughly analyzed, as well as their evolution with other parameters: time, sediment grain size,...

We thus hope that these velocimetric and stereometric first attempts will open up new ways for further analysis and modelling of the complex phenomena of antidunes.

Part V

Appendices

Appendix A

Pictures of natural bedforms

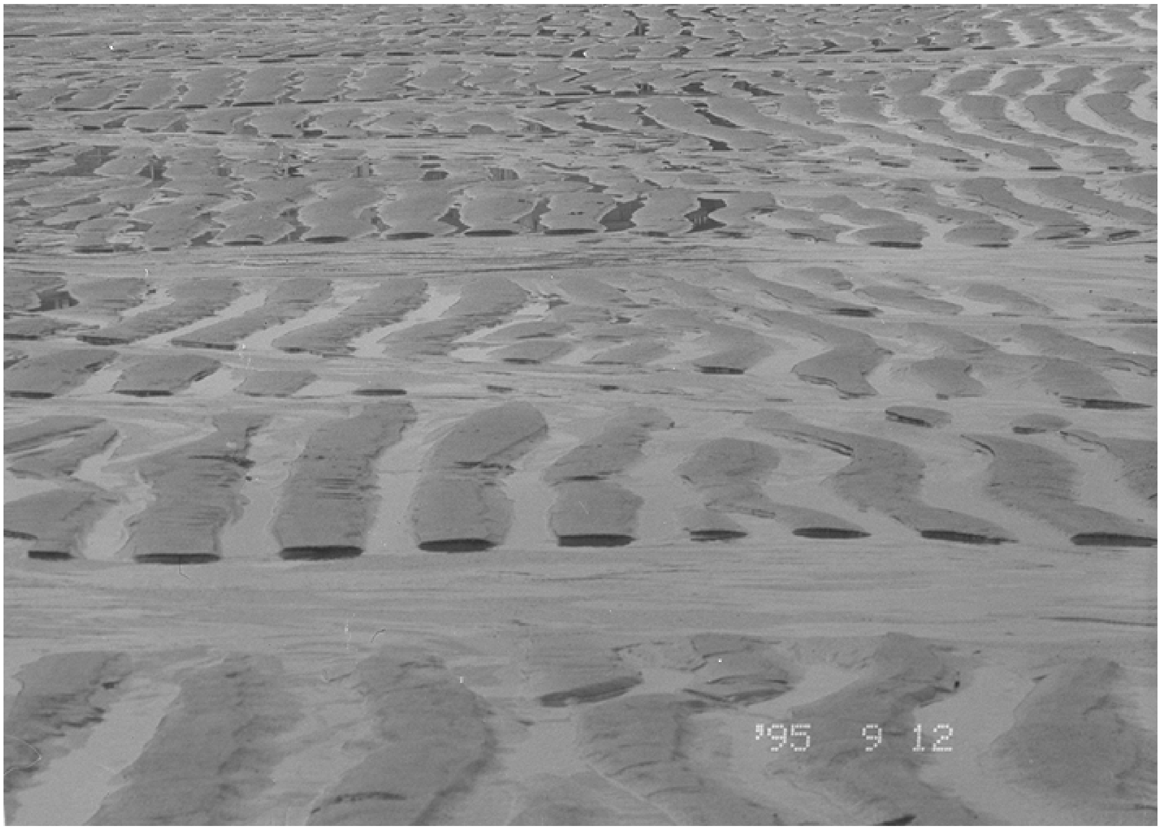


Figure A.1: The well-known oscillating flow ripples on the beach.

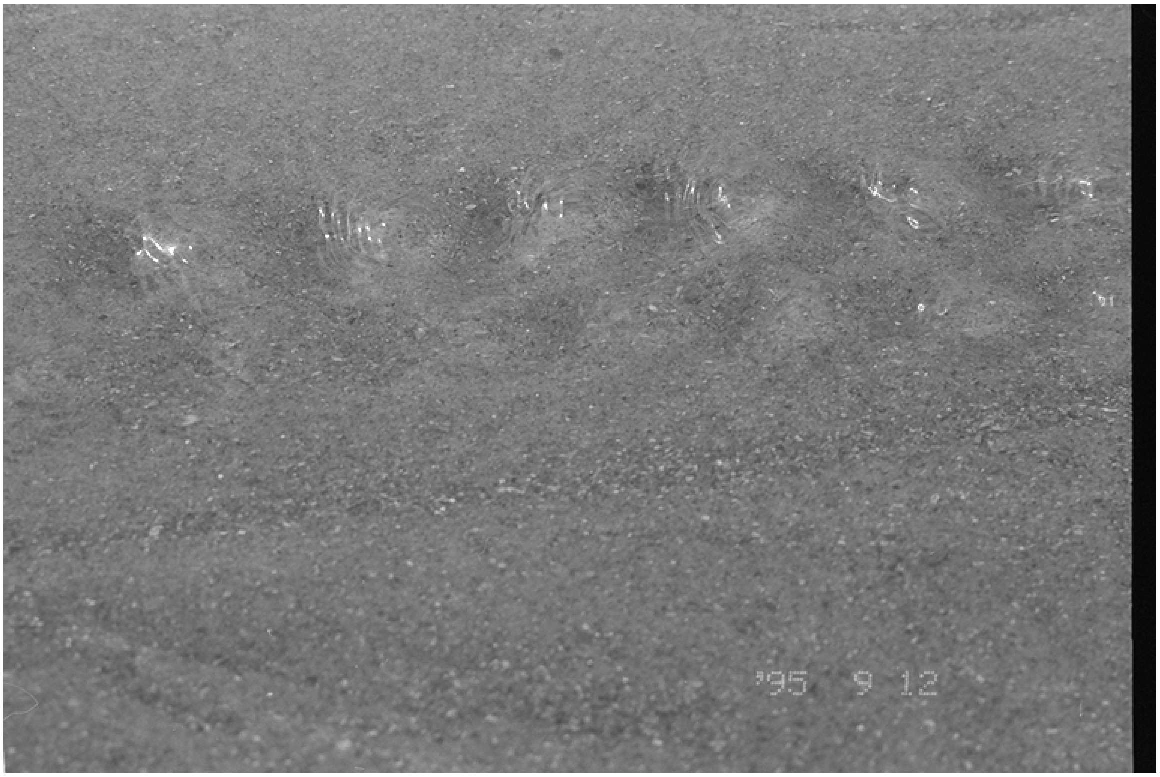


Figure A.2: Peaky patterns, like run 4.

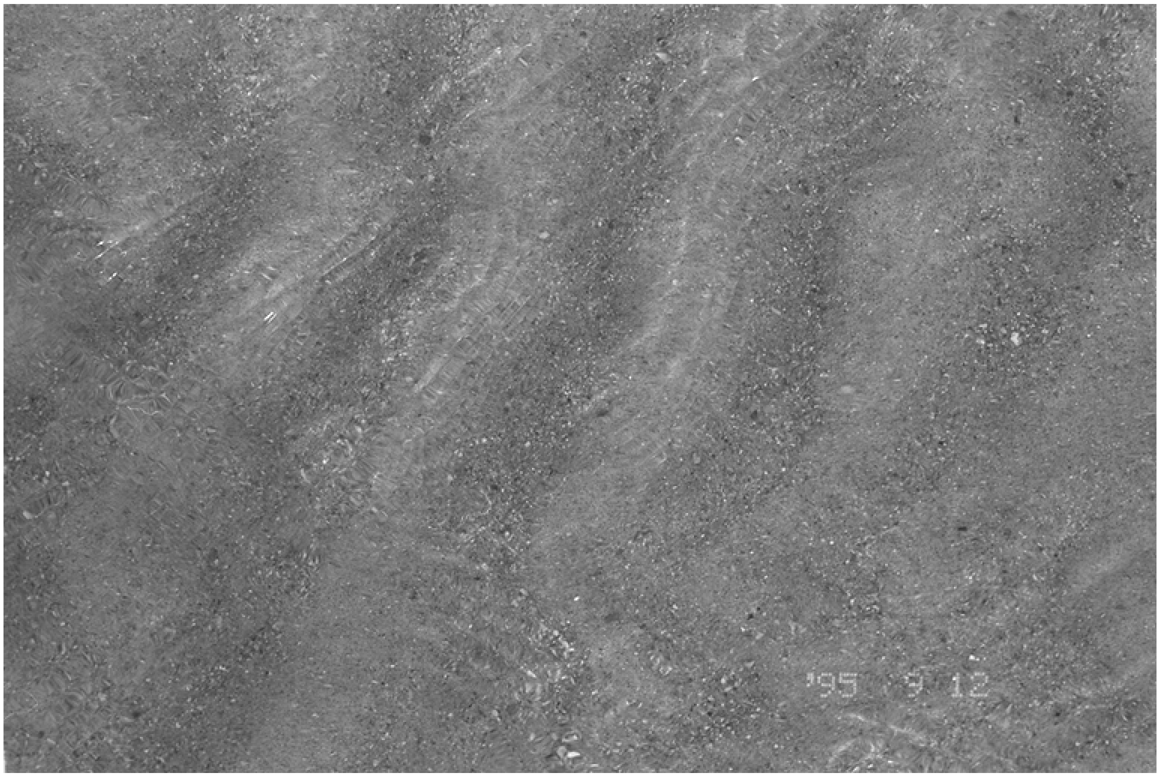


Figure A.3: Unidimensional rolls, to be compared with run 3.

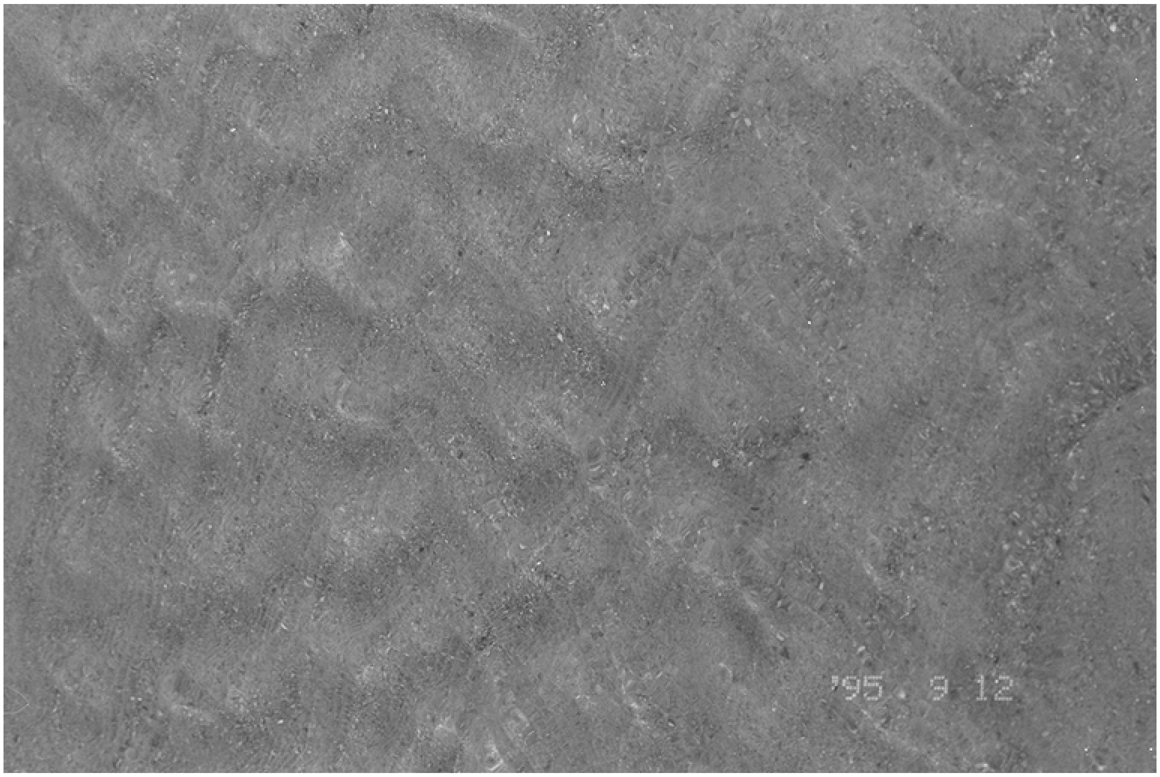


Figure A.4: A complex pattern, nearly matching run 6.

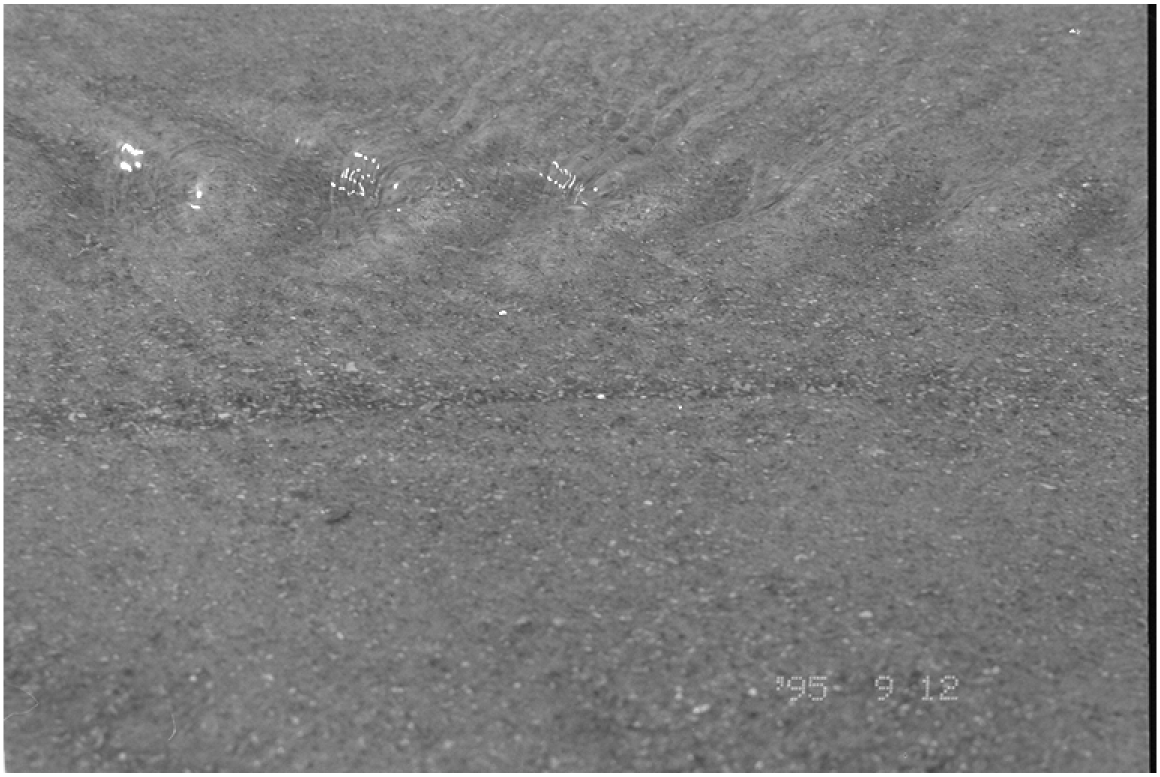


Figure A.5: The primitive V-shaped pattern of fig. A.4.

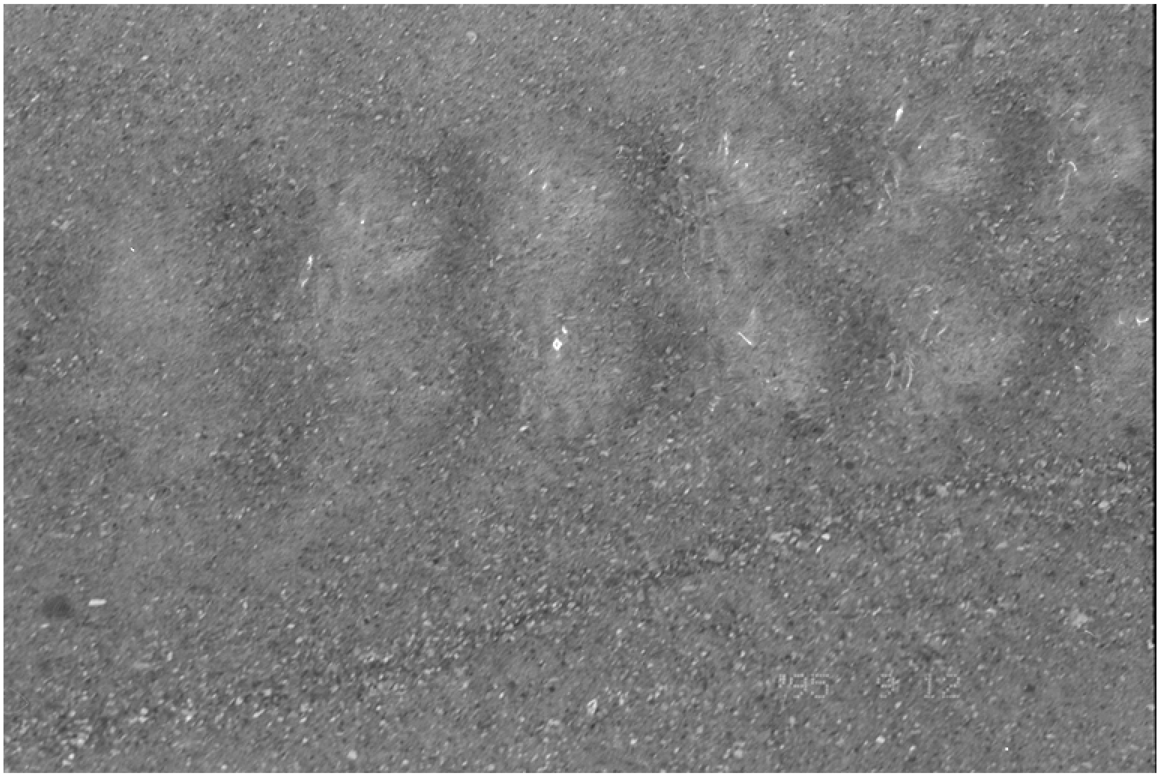


Figure A.6: A changing pattern, from rolls to peaks.

Appendix B

Pictures our experimental setup

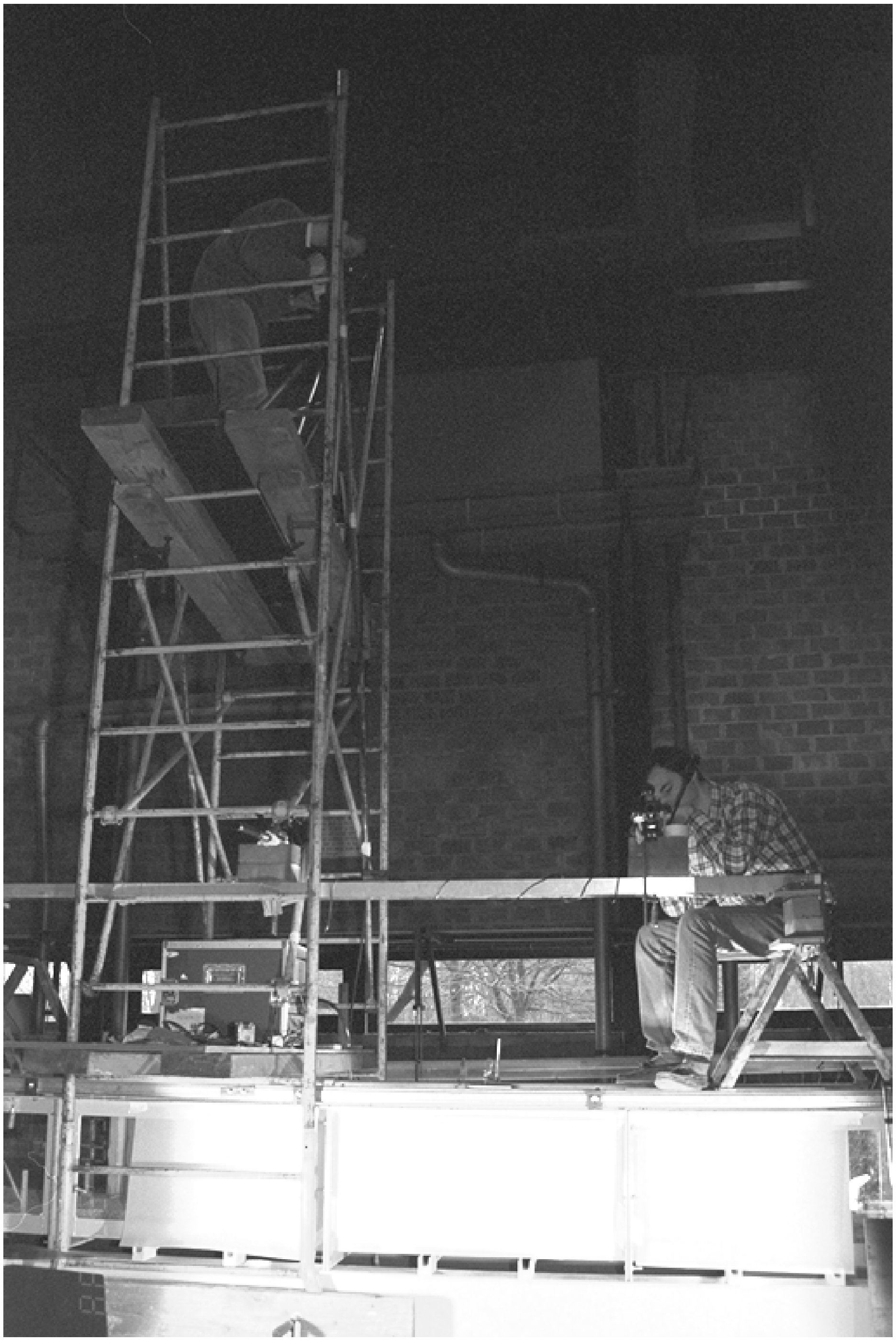


Figure B.1: The complete setup.

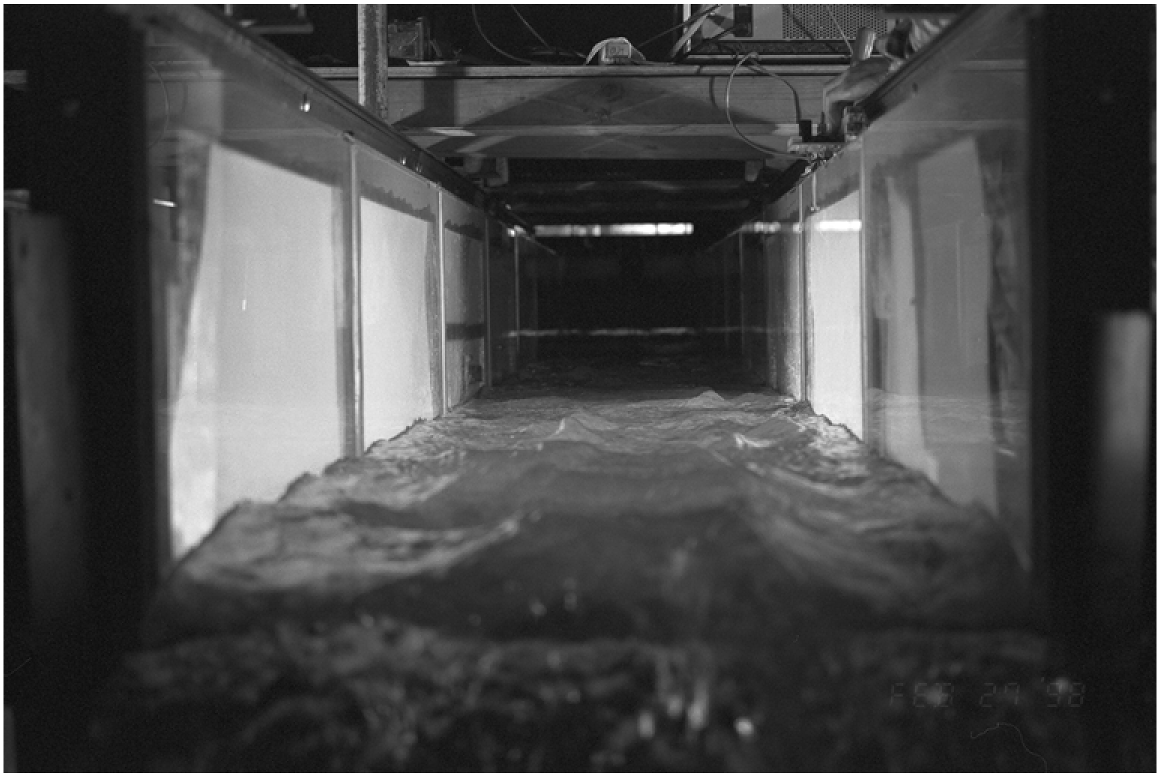


Figure B.2: A front view of the flume.

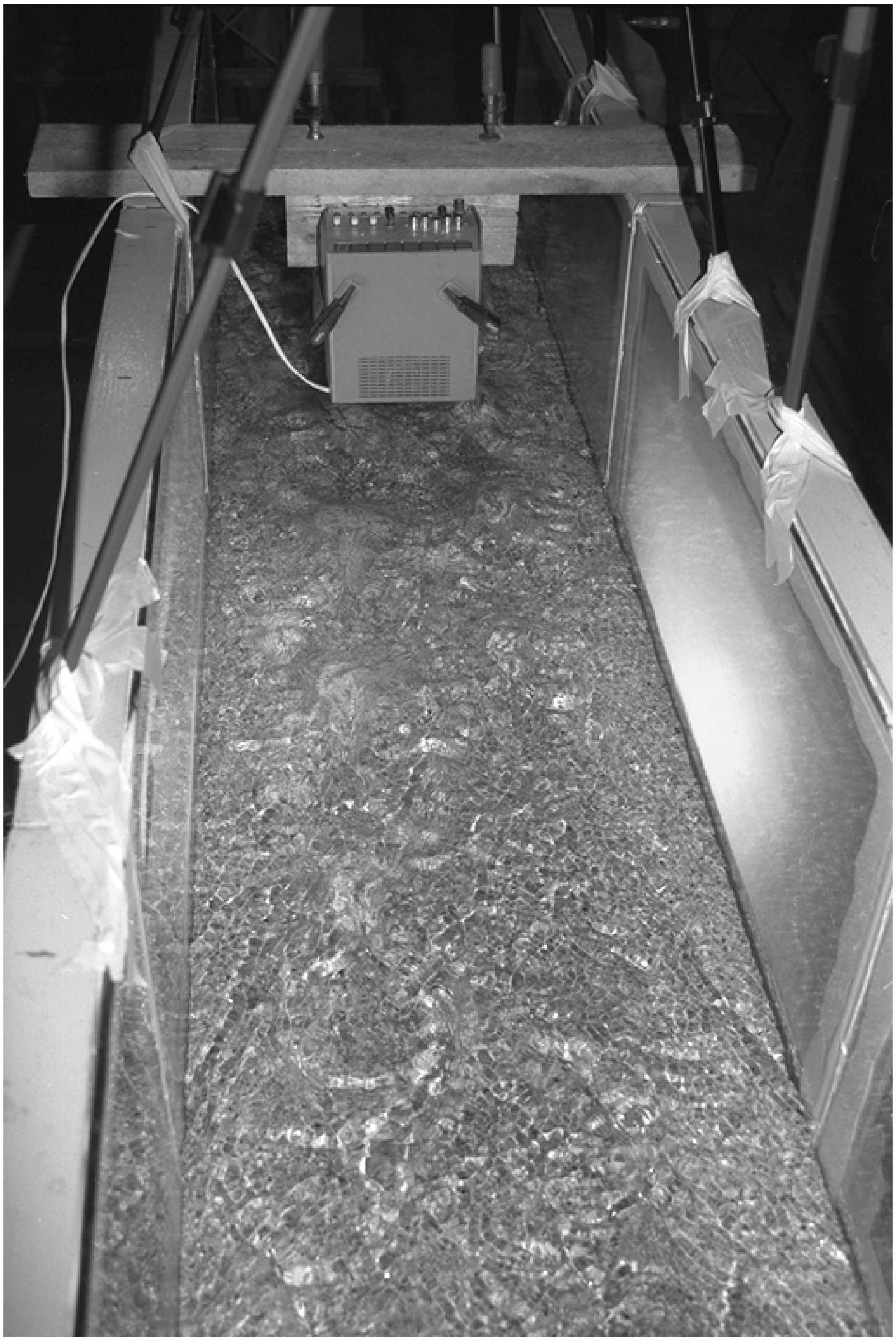


Figure B.3: A view from the flow including the first counter we used.

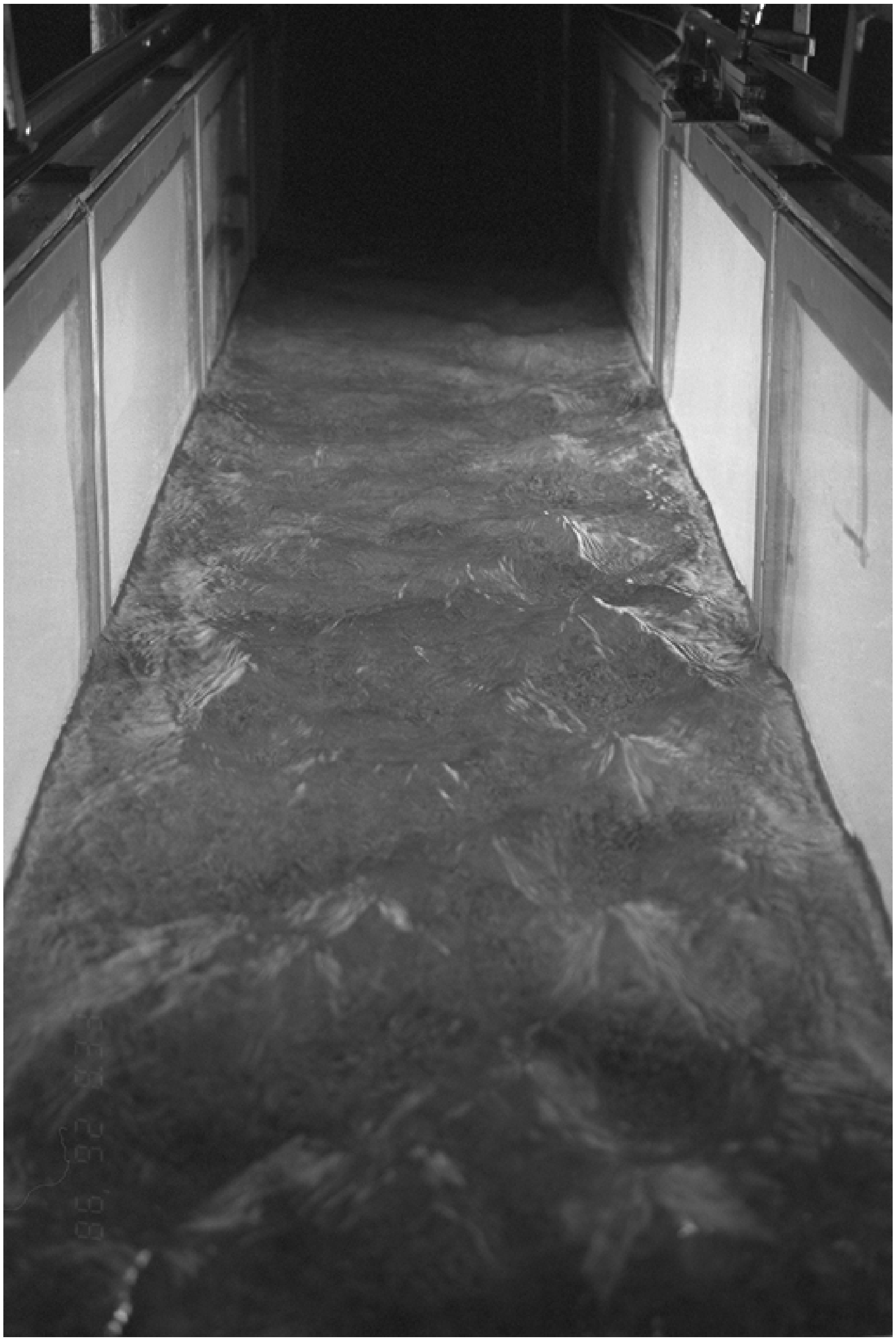


Figure B.4: Another view of the flow.

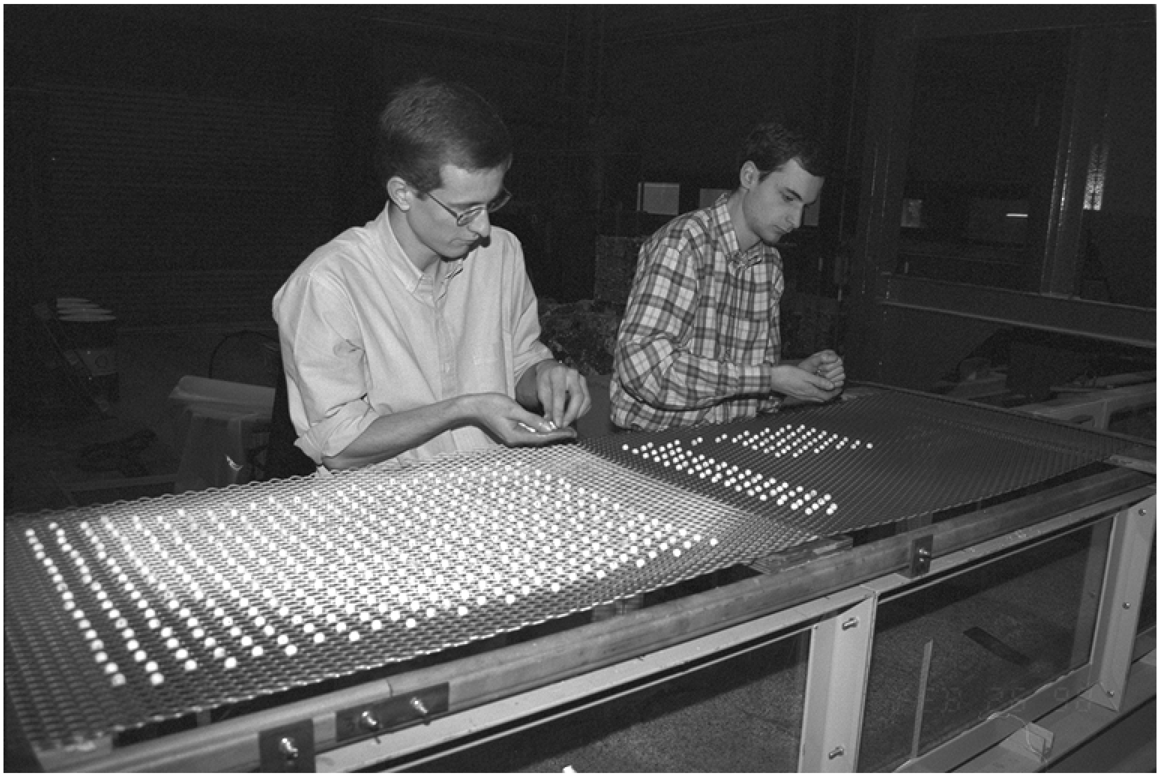


Figure B.5: The laborious task of the particles grid refitting: Dimitri Devriendt and Damien Douxchamps.

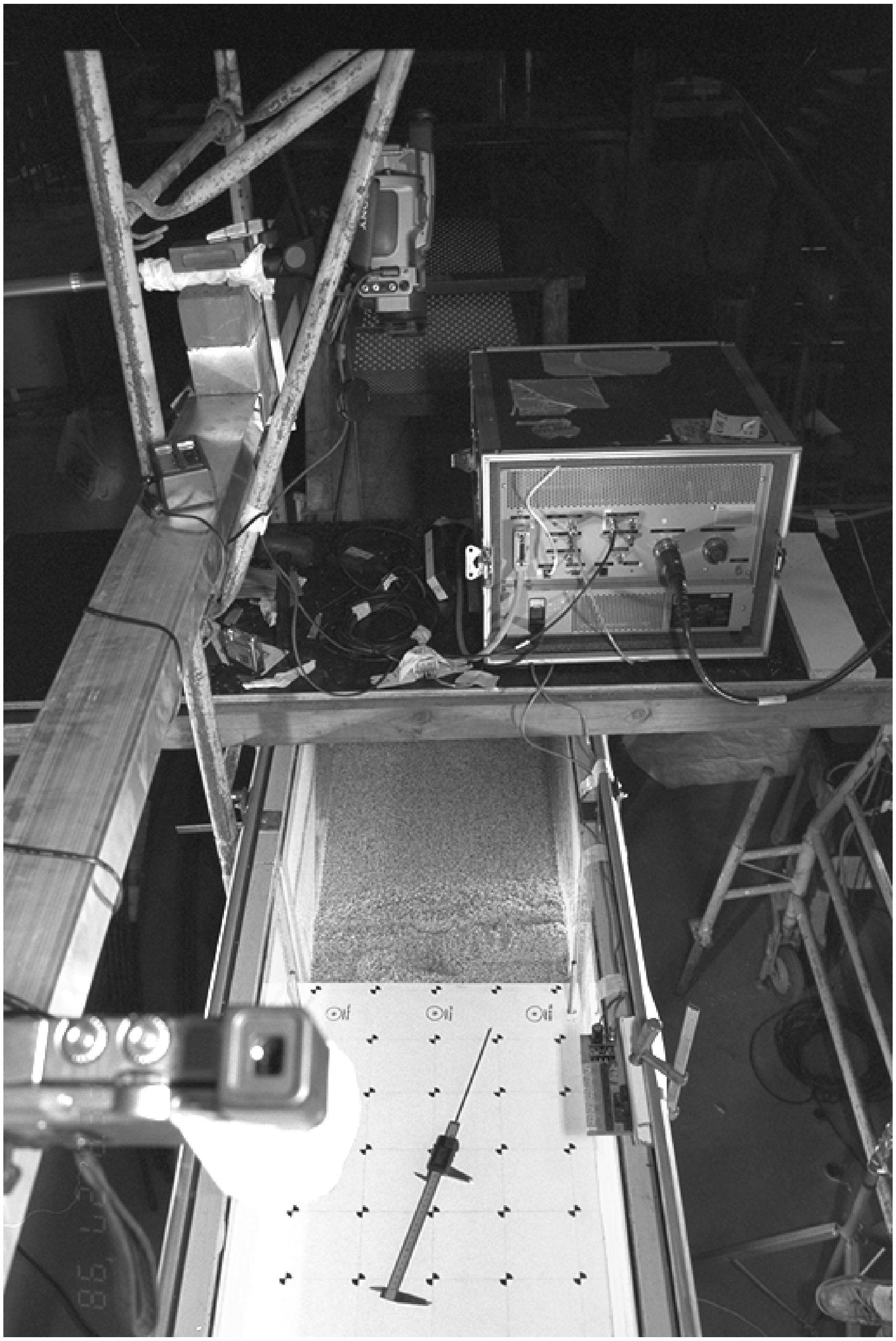


Figure B.6: The stereo setup, during the calibration phase.

Bibliography

[1] R. Jain, R. Kasturi and B. Shunck, **Machine Vision**, McGraw-Hill, 1995.

[2] J.F. Kennedy, **The Formation of Sediment Ripples, Dunes, and Antidunes**, Annual Review of Fluid Mechanics, Vol.1, 1969, pp 147-166.

[3] A.J. Reynolds, **Waves on the Erodible Bed of an Open Channel**, J.Fluid Mech., Vol.22, part 1, 1965, pp 113-133.

[4] J.R.L. Allen, **Current Ripples, their relation to patterns of water and sediment motion**, North-Holland Publishing Company, 1968.

[5] R.J. Adrian, **Particle-Imaging Techniques for Experimental Fluid Mechanics**, Annual Review Fluid Mechanics, vol.23, 1991, pp 261-304.

[6] D. Devriendt, D. Douxchamps, H. Capart, C. Craeye, B. Macq, Y. Zech, **Three-dimensional reconstruction of a periodic free surface from digital imaging measurements**, Advances in Fluid Mechanics II (Eds. M. Rahman, G. Comini, C.A. Brebbia), International Series on Advances in Fluid Mechanics, Vol.21, pp 203-212, Computational Mechanics Publications.

[7] D.Devriendt, **Reconstruction tridimensionnelle de la surface d'un écoulement surnti-dunes à partir de ses lignes de courant acquises par imagerie digitale**, Bachelor Thesis, June 1998, Louvain-la-Neuve.

[8] M. Bellal, Personal communication, May 1998.

[9] R.N. Strickland, H.I. Hahn, **Wavelet Transform for Object Detection and Recovery**, IEEE Trans. Image Proc., Vol.6, No.5, May 1997, pp 724-735.

[10] S. Mallat, **A Theory for Multiresolution Signal Decomposition: The Wavelet Representation**, IEEE Trans. Pat. Anal. Mach. Intell., Vol.11, No.7, July 1989, pp 674-693.

[11] I. Daubechies, **Orthonormal bases of compactly supported wavelets**, Comm. Pure Appl. Math., Vol.41, Nov.1988, pp. 909-996.

[12] H. Capart, O. L. Young, Y. Zech, **Voronoï-based Techniques for the digital imaging caracterization of rapid granular flows**, in preparation.

[13] R. Lengagne, O. Monga, **Using Crest Lines to Guide Surface Reconstruction from Stereo**, ICIP '96 Proceedings, Vol.2, pp 847-850.

[14] V. Koivunen, P. Kuosmanen, J. Astola, **Orthogonal Spline Fitting in Range Data**, ICIP '96 Proceedings, Vol.2, pp 385-388

[15] P. Pochec, **Moire Based Stereo Matching Technique**, ICIP '95 Proceedings, Vol.2, pp 370-373.

[16] K. Sato, **Range Imaging Based on Moving Pattern Light and Spatio-Temporal Matched Filter**, ICIP '96 Proceedings, Vol.1, pp 33-36.

[17] S.H. Ryu, T.E. Kim, J.S. Choi, **Shape reconstruction Using Estimated Surface Reflectance Properties**, ICIP '96 Proceedings, Vol.1, pp 29-32.

[18] P. Lavoie, D. Ionescu, E. Petriu, **3D Reconstruction Using an Uncalibrated Stereo Pair of Encoded Images**, ICIP '96 Proceedings, Vol.2, pp 859-862.

[19] W. Grimson, **Computational experiments with a feature based stereo algorithm**, IEEE Trans. Pat. Anal. Mach. Intell., Vol.7, Jan.1985, pp 17-34.

[20] R. O. Duda, P. E. Hart, **Pattern Classification and Scene Analysis**, Wiley, New York, 1973.

[21] D.V. Papadimitriou, T.J. Dennis, **Epipolar Line Estimation and Rectification for Stereo Image Pairs**, IEEE Trans. Image Proc., Vol.5, No.4, Apr. 1996, pp 672-676.

[22] R. Haralick, S. Sternberg, Zhuang X., **Image Analysis Using Mathematical Morphology**, IEEE Trans. Pat. Anal. Mach. Intell., Vol.9, No.4, July 1987, pp 532-550.

[23] T. Eltoft, R.J.P. deFigueiredo, **Illumination Control as a Means of Enhancing Image Features in Active Vision Systems**, IEEE Trans. Image Proc., Vol.4, No.11, Nov. 1995, pp 1520-1530.

[24] S. Ghosal, R. Mehrotra, **A Moment-Based Unified Approach to Image Feature Detection**, Trans. Image Proc., Vol.6, No.6, June 1997.

[25] N. L. Chang, A. Zakhor, **View Generation for Three-Dimensional Scenes from Video Sequences**, Trans. Image Proc., Vol.6, No.4, April 1997.

[26] J. Alexander, C. Fielding, **Gravel Antidunes in the Tropical Burdekin River, Queensland, Australia**, Sedimentology, Vol.44, 1997, pp.327-337.

[27] R.E Baddour, S.W. Song, **Periodic water waves on shear currents**, Advances in Fluid Mechanics II (Eds. M. Rahman, G. Comini, C.A. Brebbia), International Series on Advances in Fluid Mechanics, Vol.21, pp 213-222, Computational Mechanics Publications.