Grid computing for the numerical reconstruction of digital holograms

JJ Nebrensky*, PR Hobson, PC Fryer†

Department of Electronic and Computer Engineering, Brunel University
Uxbridge, Middlesex UB8 3PH, England

ABSTRACT

Digital holography has the potential to greatly extend holography's applications and move it from the lab into the field: a single CCD or other solid-state sensor can capture any number of holograms while numerical reconstruction within a computer eliminates the need for chemical processing and readily allows further processing and visualisation of the holographic image. The steady increase in sensor pixel count and resolution leads to the possibilities of larger sample volumes and of higher spatial resolution sampling, enabling the practical use of digital off-axis holography.

However this increase in pixel count also drives a corresponding expansion of the computational effort needed to numerically reconstruct such holograms to an extent where the reconstruction process for a single depth slice takes significantly longer than the capture process for each single hologram. Grid computing - a recent innovation in large-scale distributed processing - provides a convenient means of harnessing significant computing resources in an ad-hoc fashion that might match the field deployment of a holographic instrument.

In this paper we consider the computational needs of digital holography and discuss the deployment of numerical reconstruction software over an existing Grid testbed. The analysis of marine organisms is used as an exemplar for work flow and job execution of in-line digital holography.

Keywords: holography, digital holography, numerical reconstruction, Grid computing, data extraction

1. INTRODUCTION

Digital holography has the potential to greatly extend holography's applications and move it from the lab into the field: a single CCD or other solid-state sensor can capture any number of holograms while numerical reconstruction within a computer eliminates the need for chemical processing and readily allows further processing and visualisation of the holographic image. The steady increase in sensor pixel count and resolution leads to the possibilities of larger sample volumes and of higher spatial resolution sampling, enabling the practical use of digital off-axis holography.

However this increase in pixel count also drives a corresponding expansion of the computational effort needed to numerically reconstruct such holograms to an extent where the reconstruction process for a single depth slice takes significantly longer than the capture process for each single hologram. Grid computing - a recent innovation in large-scale distributed processing - provides a convenient means of harnessing significant computing resources in an ad-hoc fashion that might match the field deployment of a holographic instrument.

In this paper we consider the computational needs of digital holography and discuss the deployment of numerical reconstruction software over an existing Grid testbed. Instead of concentrating on the technical details, however, this paper will describe a particular end application – the analysis of marine particulates – and attempt to explain how the approaches to digital holography and Grid computing that we have chosen relate to actual problems encountered in building and deploying a field instrument.

---

* E-mail: henry.nebrensky@physics.org

† Now with: Van Aaken Developments Ltd., Telford Avenue, Crowthorne, Berkshire, RG45 6XA, UK

Photonics Applications in Astronomy, Communications, Industry, and High-Energy
Physics Experiments III, edited by Ryszard S. Romaniuk, Proc. of SPIE Vol. 5775
(SPIE, Bellingham, WA, 2005) · 0277-786X/05/$15 · doi: 10.1117/12.610677

2. 'HOLOMAR' – AN EXAMPLE OF HOLOGRAPHY IN ACTION

2.1. Background

Understanding the behaviour of marine biological communities is crucial to understanding the global environment. This includes biological interactions: for example, phytoplankton are eaten by zooplankton which, in turn, sustain fish; how many of each class of fish can be caught before a boom in the zooplankton population wipes out the phytoplankton, collapsing the whole food chain? Interactions with the atmosphere are also important: how will the phytoplankton population react to increased levels of CO_2 in the air, and could this help to control global warming? There are also chemical cycles to consider: an increase in nutrient levels caused by agricultural fertiliser run-off or deliberate seeding will result in more phytoplankton; will this be followed by a rise in the number of zooplankton and so fish, and thus help support fragile fisheries, or will the zooplankton be overwhelmed, leading to a toxic bloom and eutrophication?

The modelling of such processes depends on the study of aggregates of biotic or abiotic particles in order to gain knowledge of their distribution and interaction dynamics: it is necessary to understand the inter-relationships between various organisms. However, the accuracy of such measurements has been limited by the absence of good measurement techniques. Aggregates forming floc or "marine snow" can be very complex and extremely frail, and marine particles in general vary in size from sub-micron to several millimetres. Typical techniques depend on sampling bottles or counters, which destroy all spatial relationships between particles, or stereo-photography or videography, which suffer from a limited sampling volume because of the limited depth-of-field of high magnification lenses[1].

Holography thus offers significant benefits for marine biology: it can record live species in their natural environment, and it can record large volumes of the water column in one short exposure. It combines true three-dimensional imaging of organisms with high image resolution over a large depth-of-field and a wide recording dynamic range. When combined with suitable analysis techniques, holography can thus provide measurements of the size and relative position *in situ* of marine organisms coupled with species identification and classification at genus level, from which can be extracted the quantitative distribution of organisms, and thus an understanding of their spatial inter-relationships.

Studies of plankton therefore begin early in the history of holography, with the work of Knox with still images[2] and holographic movies[3], and a number of subsea holocameras looking at both plankton and sedimentary particles have been deployed; e.g. those of Stewart *et al.*[4], Costello *et al.*[5], Katz *et al.*[6], Owen and Zozulya[7] and the HoloMar collaboration (Watson *et al.*[8]) which will be described below. A detailed review of the use of holography in marine biology is given by Hobson and Watson[9].

2.2. HoloMar – an actual application of holography

We next describe the holocamera and hologram replay and analysis system developed by the HoloMar collaboration. The aim here is not to provide a technical description of a scientific instrument, but merely to provide background on the requirements and shortcomings relevant to the discussion below.

The aim of the HoloMar collaboration was to design, build and deploy a system able to provide biologists with quantitative information about the spatial distributions and interactions of marine plankton. The system may conveniently be split into three main components: a holocamera, a replay system and automated object tracking and classification software.

The underwater holographic camera "HoloCam" is able to holographically record large volumes of the upper water column containing marine plankton and seston[8,10,11] at depths of down to 100 m. It uses both the in-line (Gabor) and off-axis (Leith-Upatnieks) geometries, though the latter is not relevant to the current paper. The in-line geometry records a water volume of 92 mm diameter by 470 mm long (about 3 500 cm^3 volume). Holograms are taken using a frequency-doubled pulsed Nd:YAG laser (producing about 700 mJ per pulse at 532 nm) built inside the camera housing. The holograms are recorded on Agfa Millimask emulsion on 10 cm square glass plates. Obviously it is impractical to have to recover "HoloCam" after every exposure to change the plate, so for each geometry there is a plateholder system consisting of a light-tight cassette holding approximately twenty plates and a grabbing mechanism that can remove one plate at a time from the cassette for exposure and then replace it before taking out another. After removal from the holocamera the exposed plates must be returned to shore before being developed and fixed. The overall dimensions of the camera housing are approximately 2.4 m long by 1 m in diameter and HoloCam weighs 2.3 tonnes in air when fully loaded and ballasted.

Figure 1 shows "HoloCam" being lowered into Loch Etive (Scotland) during a test deployment. The cylindrical body of the holocamera contains the laser, power supplies, beam expansion optics, control electronics and plateholder cassettes. The box on the right-hand end contains the turning mirror and output window for the in-line beam travelling diametrically past the camera end to the plateholder exposure window opposite (just visible).

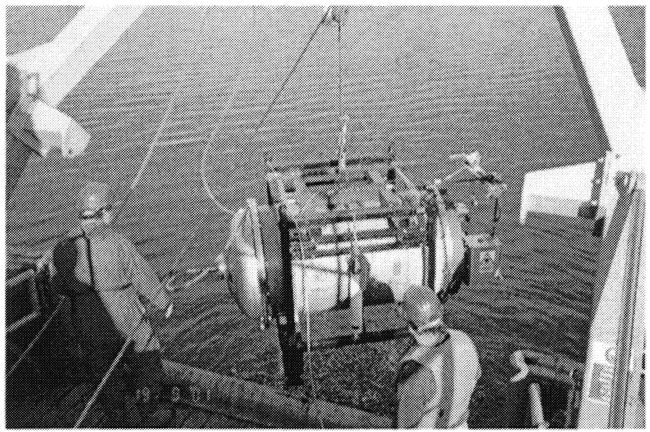

Figure 1: Lowering of HoloCam into Loch Etive (from [8])

By themselves the artefacts produced by the holocamera – opaque glass plates – are not very useful: it is necessary to use a second laser to illuminate the developed plates to reconstruct (in air) the holographic real image of the sample volume; by scanning a videocamera with magnifying optics through this volume it is possible to find and locate the organisms within that 3-d space[11]. The HoloMar collaboration has therefore also built a holographic replay system "HoloScan" providing high-resolution reconstructed images (figure 2) from a scanable volume of 25 x 25 x 1000 mm[12,13]. Organisms and aggregate particles can be identified visually, and their position found from the position of the videocamera (calibrated against images of fiducial wires attached to HoloCam).

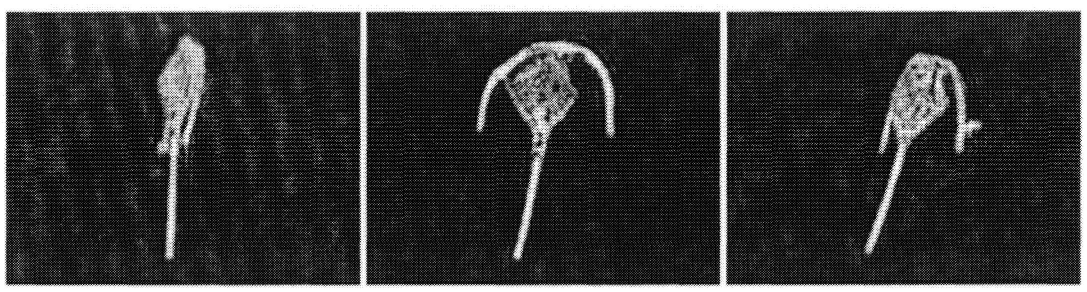

Figure 2: Sample images from in-line holograms of plankton: *Ceratium Tripos* (each image is 430μm wide) [12]

Although casually finding and identifying organisms manually is quite easy, the reliable extraction of quantitative data is a non-trivial exercise: the analysis time for each plate can be several person-weeks. The issues with such protracted data extraction are not merely the time scale, but that with any manual involvement operator fatigue can result in the introduction of hidden systematic errors in the results. The need for automated analysis has therefore been addressed by coupling the video output from HoloScan with object tracking and extraction software, which searches through successive video frames to find the plane of best focus of a particle and then extracts and binarises the object sub-image[12], and passes it to a neural-net based classifier for identification[13]. The replay system thus allows identification of species, size, relative location and distribution of marine organisms without operator intervention. Even then it is a non-trivial undertaking: using a brute force approach with HoloScan set to high magnification (one CCIR frame images an area 1.0 by 0.7 mm) and taking 0.1 mm steps in depth (larger than many objects of interest), an in-line sample volume

100 mm diameter and 500 mm long would yield about 30 Terabytes of raw data to search through; it would need at least 2 weeks to scan in, simply due to the CCIR frame rate of 25 Hz. Of course, the information content of such holograms is much lower: Royer's criterion for reconstruction of in-line holograms requires that for good replay at least 95% of the recording beam pass through the sample volume undisturbed, which means that there will be a maximum of about 200 Mb of in-focus particle images, and the storage size of biologically-relevant information (object classification, size, attitude and position) will be even smaller.

3. DIGITAL HOLOGRAPHY

3.1. Benefits

Digital holography (strictly, the digital **recording** of holograms) involves replacing the photographic material used in conventional holography with an electronic imaging sensor such as a CCD array. The fringe pattern recorded by the sensor at the moment of exposure is then transferred to a computer and reconstructed numerically (figure 3).

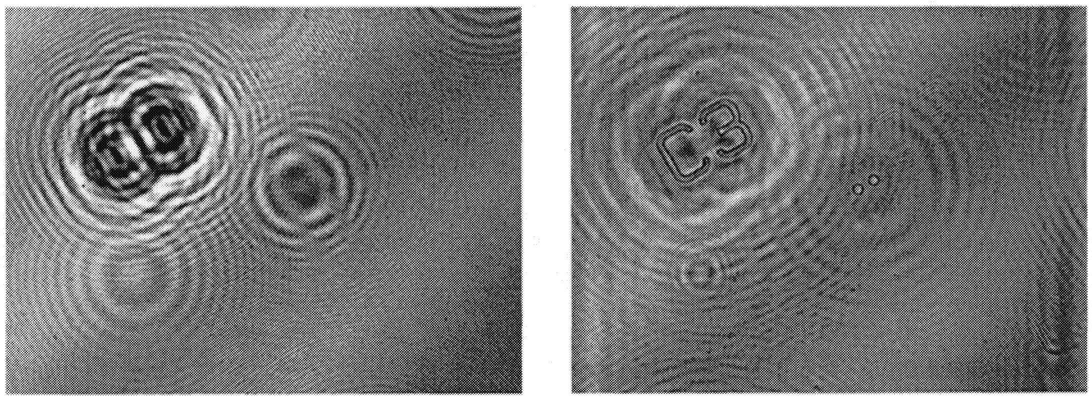

Figure 3: An in-line hologram of a test target, captured from a CCIR videocamera (*left*) and the in-focus objects, regenerated within the computer (*right*).

This avoids the need to handle glass plates within the holocamera, eliminates chemical development and can compensate for the refractive index change. The holocamera can be made both smaller (thus causing less disturbance in the water) and lighter (and so easier and cheaper to deploy, and operation possible in rougher weather). Numerical reconstruction not only makes the results available directly in digital form ready for further analysis and visualisation, but analysis can begin immediately as there is no need to return the plates to shore before processing (carrying out the chemical process in a darkroom with trays of fixer on a swaying boat is hardly a viable option). There are also further practical advantages associated with avoiding the needs to open frequently the holocamera pressure vessel in the field (to change plate cassettes) and to store used photographic chemicals for months pending on-shore disposal.

For ensemble-based applications such as particle analysis, it is generally desirable to sample as large a volume as possible for statistical purposes. For in-line holography, where the in-water path-length is limited by absorption and scattering, this means either using a small sensor with a rapid capture rate and aggregating the sample volume in time as in Sun *et al.*[14] (or even – if the framerate is high enough – building up a larger volume from overlapping sub-volumes) or using a larger area sensor – for example one manufacturer has recently announced a full-frame CCD sensor with 8.75 µm by 8.75 µm pixels over an 80 mm x 80 mm image area[15] which could thus record electronically sample volumes comparable to those recorded "photographically" by HoloCam.

3.2. The numerical reconstruction process

There are several ways of calculating the reconstructed image from a digitised hologram[7, 16, 17]. Our general approach is based on the convolution method[16, 17] and can be summarised as follows:

1. Numerically multiply the hologram by the reference wave (in the case of in-line holography the reference wave is 1+0*i*),
2. take a Fourier transform
3. and multiply the result by a transfer function based on the Rayleigh-Sommerfeld equation.
4. An inverse Fourier transform gives the reconstructed image.

This process is computationally heavy as it requires multiple 2-d Fourier transforms.

Fournier-Carrié has previously written "HoloBatch"[17], a C++ code that reconstructs single image planes from digital holograms recorded with up to 8 bits. Reconstruction parameters are read in from a text file at run time, and include not only the desired wavelength and image plane depth but also zero padding to power-of-2 dimensions and a choice of either a simple Discrete Fourier Transform (DFT) or a Cooley-Tukey based Fast Fourier Transform (FFT). The code however is tied to the Linux operating system by a number of library dependencies and has a very inefficient padding implementation based on external calls to NetPBM executables. Table 1 shows a number of comparisons of end-to-end processing time, including all image loading and padding stages. Not surprisingly, the results on a single machine, "f", show that an FFT is much quicker than a DFT even if the FFT has to be applied to a significantly larger padded image. Cross-platform comparisons are illustrative rather than rigorous because of the OS and compiler differences, but the processing time on a faster machine "g" improves roughly linearly with CPU speed as expected.

One of the present authors (PCF) has refactored the code, notably to use the FFTW library[18] and to pad hologram images "on the fly" as they load. This new version, "HoloPlay" will, in principle, be platform-independent subject to FFTW portability (the current implementation depends on a Windows-only INI-file parser; we have identified cross-platform alternatives but not yet incorporated them). Typical reconstruction times on a PC "l" comparable with "f" are also given in Table 1 and show a significant improvement for large images, confirming that FFTW is indeed a faster algorithm, though FFTW's machine profiling stage uses up the time saved in the zero-padding step. HoloPlay is significantly more memory-hungry though: even with 512 Mb in the machine there is noticeable swapping when reconstructing `8MWIRE.PGM`. Note that the timings in table 1 refer to single image planes rather than entire sample volumes, which might consist of hundreds of such slices.

Table 2 shows timings of the HoloMar object tracking code[12] applied to typical image sequences – though with certain parameters set to ensure that NO objects are successfully located – which takes about 6 s per CCIR frame. Tracking particles adds an overhead of about 2 s for those frames actually containing objects, the effect of which on the overall time will of course depend on both the type of particles the code is set to look for and the number of such particles in a particular image sequence. Taking into account foreseeable optimisations of the tracking code, we believe that image reconstruction times two to three times as long as the average object finding and tracking time are representative for our analysis chain at CCIR video sizes. How the object tracking times scale with frame size has not yet been investigated however.

Although the reconstruction process for each image plane can be carried out independently, the object tracking requires all the image planes, in depth order. While analysis times on the order of seconds with contemporary hardware ("g") don't provide much of an incentive for remote processing of CCIR frames to be sent and returned over a 100 Mbit LAN, we have pointed out above that there is strong pressure to use significantly larger frame sizes with their much greater processing times (and coincidentally the commoditisation of gigabit Ethernet has the potential to reduce transfer times). For example, we are already testing HoloPlay in the lab with a COTS camera (Atmel Camelia 8M[19]) capturing holograms of 8 Mpixels with 12-bit depth (e.g. `8MWIRE.PGM`). From table 1 it can be seen that this will take minutes to reconstruct one image plane and over an hour for a volume. An even more extreme example is the large-area sensor mentioned in section 3.1: to cover 80 mm with 9 μm pixels requires a sensor resolution of 9216 *x* 9216 – that's **85 Mpixels** – before padding[15].

The single desktop workstation thus presents a bottleneck to the processing of large volumes of holographic data, which can be avoided either by deploying specialised hardware or by spreading the load across multiple processors. As mentioned above, the reconstruction of each image plane is essentially independent from that of the others so that effective parallelisation does not depend on exact synchronisation or message passing between processes: it is possible to use a heterogeneous cluster of standard PCs. "Grid computing" is thus an attractive and viable option.

```
Image                        Transform and padding            Time (wall-clock)
-------------------------    -------------------------------- -----------------
f: P2 400MHz 384Mb 512K cache (L2, in the CPU cartridge), RedHat Linux 6.2, gcc-egcs 2.91
SAVVAS1.PGM    768x576 8bit   DFT with no padding  (0.44Mpixels)     914 seconds
SAVVAS1.PGM    768x576 8bit   DFT pad to 1024x1024 (    1Mpixels)   3305 seconds
SAVVAS1.PGM    768x576 8bit   FFT pad to 1024x1024 (    1Mpixels)     17 seconds
8MC.PGM       1024x512 8bit   FFT with no padding  ( 0.5Mpixels)       8 seconds
8MC.PGM       1024x512 8bit   DFT with no padding  ( 0.5Mpixels)    1240 seconds
8MWIRE.PGM   2300x3500 8bit   FFT pad to 4096x4096 (   16Mpixels)    671 seconds
8MWIRE.PGM   2300x3500 8bit   DFT pad to 4096x4096 (   16Mpixels) 216504 seconds
8MWIRE.PGM   2300x3500 8bit   DFT with no padding  (    8Mpixels)  73272 seconds

g: Athlon XP2600+ (1900MHz) 1024Mb 384K cache, RedHat Linux 7.3, gcc 2.95.2
                  (RH reports 512K cache, CPU spec sheet gives 128K L1 cache, 256K L2)
SAVVAS1.PGM    768x576 8bit   FFT pad to 1024x1024 (    1Mpixels)      3 seconds
8MC.PGM       1024x512 8bit   FFT with no padding  ( 0.5Mpixels)       2 seconds
8MWIRE.PGM   2300x3500 8bit   FFT pad to 4096x4096 (   16Mpixels)    101 seconds

1: P2 400MHz 768Mb 512K cache, Microsoft Windows NT4 (SP5), Visual C++ 6
SAVVAS1.PGM    768x576 8bit   FFT pad to 1024x1024 (    1Mpixels)     12 seconds
8MC.PGM       1024x512 8bit   FFT with no padding  ( 0.5Mpixels)      10 seconds
8MWIRE.PGM   2300x3500 8bit   FFT pad to 4096x4096 (   16Mpixels)    185 seconds
```

Table 1: Comparison of reconstruction times for single image plane using HoloBatch (Linux) and HoloPlay (Windows). Code compiled with default (release) optimization; no swapping observed on any platform.

```
Code and volume       Frames                                   Time (wall-clock)
--------------------- ------------                             -----------------
1: P2 400MHz 768Mb 512K cache, Microsoft Windows NT4 (SP5), Visual C++ 6
HS20120A ci6a 768x566 1-459 off HD  - finds nothing               2588 seconds
HS20120A co2a 768x574 1-61  off HD  - finds nothing                360 seconds
```

Table 2: Simplified object tracking analyses for HoloMar[12] code and image sequences, from in-line (ci6a) and off-axis (co2a) holograms.

4. INTRODUCTION TO 'THE GRID'

4.1. What is 'the Grid'?

Although there is currently widespread discussion about "Grid" computing, it is often unclear exactly the term means, especially compared to the established term "distributed computing". By "The Grid" here we mean an infrastructure that allows general purpose computing (rather than being limited to analysing extra-terrestrial emissions or finding prime numbers); supports the notion of a "Virtual Organization" (VO) that controls how its sub-set of the total Grid resources are to be shared among its members; and is geared towards wide-area deployment, with heterogeneous resources spread across the globe. This is similar to the vision originally proposed by Foster and Kesselman[20, 21] and since implemented by them as the Globus Project middleware[22].

From the user's perspective, the World-Wide Web provides seamless access to spread-out data (activate a hotlink and the page is delivered to the browser, no matter if it was on the next desk or on the next continent) – without the underlying protocols (HTTP, HTML) needing to care about who you are. Similarly, the Grid should provide seamless access to computing (the user doesn't have to care about the location of the computer running the code); however the Grid explicitly requires each user to declare their real-world identity via X.509 cryptographic certificates. Once a user has been authenticated as a member of a suitable VO they can freely access that VO's resources. Another view of the Grid is thus that it provides a single sign-on to use facilities spread across many institutions.

A number of Grids and Grid testbeds have been implemented; we refer here to the model followed by the EU DataGrid[23] project and since deployed as the LHC Computing Grid (LCG)[24]. Unfortunately the field is presently encumbered by constantly changing acronyms; hence the terminology here is a personal selection.

4.2. How does a Grid job happen?

The basic cycle involved in running a job over the Grid is shown in figure 4. At a computer that acts as a User Interface (UI), the user describes the job to be run: the executable, data files to be processed and other requirements (e.g. operating system, memory needed) are specified using JDL (Job Description Language). When the job is submitted, the UI client gathers the JDL and required files into a "sandbox" and passes this to a Resource Broker (RB), which assesses the job requirements and current state of Grid resources, and accordingly identifies the best place to send the job. The Grid has two basic types of resource. A Computing Element (CE) provides the CPUs that do the actual processing – jobs can only run on a CE. A Storage Element (SE) provides mass storage space in which programs and data files can be stored; thus for example rather than repeatedly sending identical copies of an image file in the sandbox to the RB we can instead upload the data once on to a suitable SE, and then refer to this as the data source in the JDL (see section 5.1). The RB can then try to choose a suitable CE close (in network terms) to the stored data to run the job. The Grid middleware will automatically take care of ensuring that the hologram file is available by the time the executable starts. In principle the Grid infrastructure should also take care of replicating the file to other SEs, both to spread the load and also as a backup. "Suitable resources," in this context, refers not only to physical attributes such as unused disk space left or the correct CPU architecture, but also that the resource supports the user's particular VO.

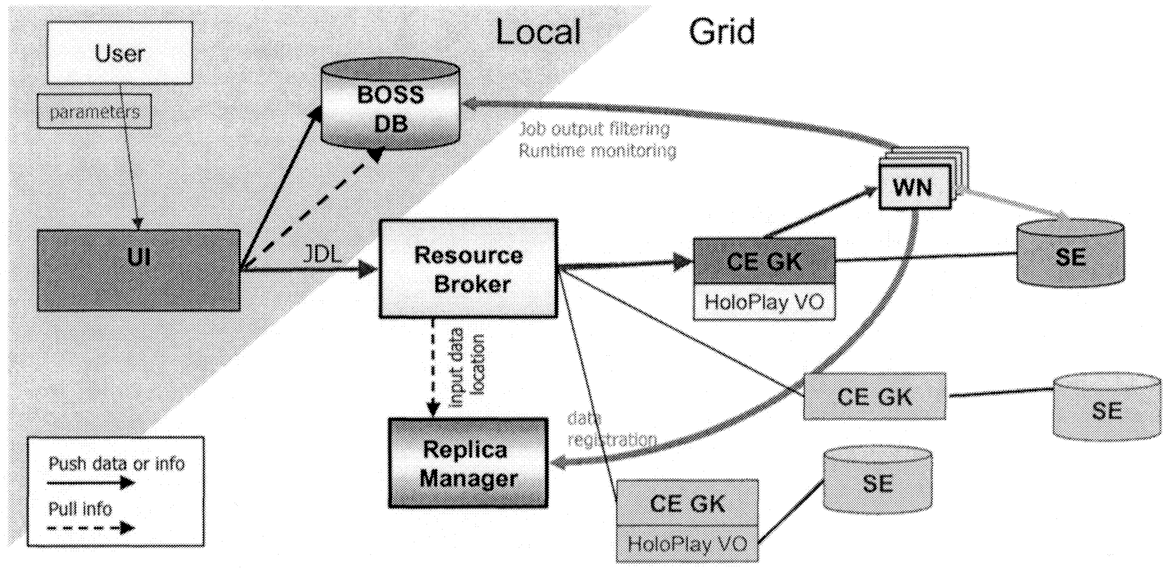

Figure 4: The Grid job cycle

A CE consists of a Gatekeeper (GK), which receives the job, and a set of Worker Nodes (WN) that do the actual work. On current Grid testbeds these are mostly many standalone Linux PCs with the gatekeeper passing each successive job to the next free WN to be run – similar to a traditional batch farm – but there is no reason why a CE should not have just one WN, which happens to be a specialised massively parallel system or similar.

The Grid infrastructure also includes many other services, such as meta-data and replica management (which relate to finding out which SEs a prticular data-file is stored on), network monitoring and various information services, but it is hoped that the end-user will not need to interact with them directly. There are a number of web portals that can provide various friendly, aggregated views of the various monitoring and status data (e.g. the monitoring maps and frameworks provided by the LCG Grid Operations Centre[25]).

Eventually the job will have been processed, and the files to be returned are collected by the CE into an output sandbox and sent back to the RB. Whenever the user desires they can be retrieved by the UI and stored on the local PC.

Since, as mentioned above, many current Grid resources are effectively batch farms, why not just use a dedicated farm or supercomputer? Obviously, dedicated resources will be expensive, because of both the capital cost and the running costs: space, power, air conditioning; but then the resources on the Grid must also be paid for somehow and it is likely that heavy Grid users will end up having to pay for their usage, if not in cash then by sharing their own paid-for resources with the rest of their VO.

Unlike with applications such as particle-physics analysis or bioinformatics where the aim is to keep a continuous throughput, dedicated resources for instruments such as a digital equivalent to HoloCam are very wasteful as they will be idle for much of time (e.g. while the ship is at sea) and then be used intensively when staff and raw data return. In the meantime, similar computing resources may be idle at another institution because the staff and equipment there are in the field... Although sharing such resources seems an obvious solution it has traditionally been difficult not only because of the scheduling, but also from the practical perspective of filling in forms to get accounts at remote sites, remembering multiple login IDs and passwords, and having to FTP data around and then telnet to each remote machine to actually run the job, and so on. In contrast, groups wishing to pool resources now only need to form a VO; the Grid then naturally provides ad-hoc access to shared resources. The key is not that the Grid makes it easy to "scavenge" idle resources, but rather that it makes cooperation to avoid wasting them in the first place much easier.

4.3. Job tracking and applications monitoring

Although the process outlined above is fairly straightforward when the user has only one or two jobs to run, it becomes overwhelming when the user must manage a large number of jobs, each a sub-task of a greater whole. For example, in the 100 slices case above we have both a need for job tracking and management (Have all needed slices been requested? Which returning sandbox contains which depth slice?) and a need for applications monitoring (are all 100 jobs running correctly, or have some crashed or hung?). Hence large-scale users of the Grid, such as particle-physics experiments, are commonly developing their own higher-level applications that sit on top of the standard middleware; however these are mostly specifically designed around the data and work-flow of a particular project and thus not directly usable for holographic applications.

We have identified a package, BOSS[26], which we believe can provide the backbone of a simple tracking and monitoring system for digital holography. BOSS essentially provides an extra layer around the UI that, as each job is submitted, stores all the relevant information about that job in a database close to the user, figure 4. The JDL and sandbox are then passed on to the Grid as before, except that a BOSS wrapper is put around the executable. When the program runs at the WN, this wrapper intercepts the standard output and error streams – which are normally redirected to a disk file and returned in the sandbox – and inserts a copy of certain entries directly into the database. Thus by querying the database it is possible to both identify the results from previous jobs and to monitor the current status of those still running. Although currently used versions of BOSS still require each wrapper at the WNs to make a remote connection into the database, an improved version is being tested which uses a standard Grid monitoring and information service (R-GMA) as a transport layer[27].

An illustration of how BOSS might be used for monitoring holographic reconstruction is given in figure 5. The user identifies a task, such as reconstructing a particular region of the sample volume, and breaks it down into a set of jobs to be run over the Grid. For each job, BOSS stores information about it in the database and then submits it to the Grid.

Eventually each job arrives at a WN and runs, updating the database with its progress, until it finishes and the data is recovered by the UI and stored somewhere on the disk. Its location can be identified from the information in the database (it is based on the partly random and very non-memorable Grid job ID number). The object tracker can then use the database to locate the next image plane to be analysed.

Gridifying the analysis chain described in sections 2 and 3 should thus be achievable simply by creating a relatively simple shell script to generate the JDL file corresponding to each successive depth plane and submit them to BOSS, and adding an interface to the tracking software so that it can query the BOSS database to find out if the next depth plane is ready, and if so whence to load the data.

So far we have considered the Grid in terms of our simple approach of reconstructing many image planes individually but at the same time, then gathering them together and letting a single instance of the object finder and tracker work its way through the "stack" of images. We have chosen this simple strategy mainly because it fits well with both the way our current HoloPlay code works and with the *modus operandi* of the Grid. In the next section we look at some other strategies.

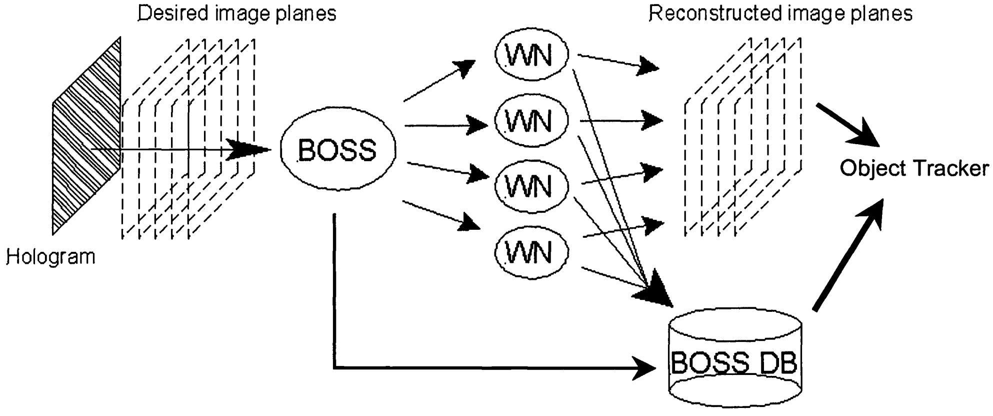

Figure 5: Data tracking and monitoring with BOSS.

5. HOW TO DISTRIBUTE RECONSTRUCTION AND IMAGE EXTRACTION JOBS ON A GRID

5.1. Reconstruction

To fully reconstruct a digital hologram the recorded hologram image must be pre-processed (dark frame subtraction and possibly flat-fielding) and then transformed for every different depth plane desired. In many applications experience has shown that, in order to correctly locate sharp images of small objects close to the recording plane (high NA), slices every 0.2 or 0.1 mm in depth are desirable. From the point of view purely of reconstructing a total image volume then every depth plane is completely independent and the job of distributing the images across WNs at multiple CEs on a Grid is then determined by the relative cost of the connection bandwidth versus the computational power advantage provided by large numbers of WNs. For example, a typical current digital hologram has a file size of order 10 Mb. If this is an in-line recording of a depth of 0.5 m and it is desired to reconstruct this at 0.1 mm depth slices then potentially the total data bandwidth needed is of order of 50 Gb. Clearly if a job distribution system can send one copy of the data file to an SE near each cluster of WNs at a Grid site then this can be very significantly reduced. In the near future CCD arrays (or individual CCD chips) which will produce hologram files of order 200 Mb in size are likely (see section 3.2) in which case the worst case (naive) job submission system would result in the transfer of up to 1 Tb of data.

It should also be noted that the initial multiplication and Fourier transform (steps 1 and 2 in section 3.2) will give the same results for all depth slices, hence the reconstruction process can be sped up by distributing the Fourier transform of the hologram instead of the hologram itself. (Intriguingly, in certain optical systems the object images have been found to contain regular fringes that must be removed with a stop-band filter[12] raising the possibility that, for some datasets, including the final inverse transform in the reconstruction stage may also be counter-productive!)

5.2. Image Extraction

Processing the primary hologram to produce a 3-d image set is only part of the story. Indeed many technically successful conventionally recorded hologram projects have not gone beyond proof-of-principle because of the sheer volume of data that needed to be processed to extract useful images[5]. So far our assumption has been that only the reconstruction will be done by the Grid while the object tracking and extraction is done centrally, but we have seen in section 3.2 that the object tracking is only a few times faster than the reconstruction step implying that this process also needs to be distributed.

In processing the image data, the first requirement is to locate the best-focus image plane for each object. Thus one needs to deal with a set of successive depth slices in order to extract the objects. However for small objects (those recorded at tens of far-field distances in in-line holography), there is only short-range correlation in depth. Thus a reconstruction of a complete volume (of order thousands of slices in depth) can be split into sub-groups providing that each sub-group has a set of depth-correlated slices. Some overlap of each of the sub-groups is needed since the final out-of-focus slice before best focus may occur in the last slice of a particular group, thus the next group must start at least at some slices prior to the last slice of the previous group. In practice, for typical objects being reconstructed in in-line holograms an overlap in depth of order 1 mm (5 or 10 slices) is needed.

The diagram on the left of figure 6 shows one possible implementation scheme: the original hologram (not shown) is uploaded to an SE and WNs in a nearby CE independently reconstruct single (or small sets of) image planes and store them also on the SE (because of other jobs at the CE and as the WNs at a site may not actually be identical, it can't be assumed that jobs submitted sequentially will complete in that order). Whenever a suitably long, contiguous sub-group becomes available, object tracking and extraction can be done on another WN. Details of any object found (indicated by the dot) are returned to the UI. This approach has the advantage that the intermediate images are in persistent storage on the SE and can readily form a single coherent volume, and can thus readily be re-analysed if desired. The difficulty lies in efficiently starting the object tracker when enough slices are available: although the information needed may in principle be available e.g. from the BOSS database, there may be a significant delay between the submission of the tracking job at the UI and its arrival at the WN.

The right-hand side of figure 6 illustrates an alternative approach: each Grid job reconstructs a sub-group, storing it in scratch space on the WN, and then searches it for objects. This has the advantage that tracking can begin immediately the sub-volume is ready and that there is no interdependence between WNs, so jobs can run at any site, rather than being tethered to a particular SE. As there is no way to force a job to run on a particular WN, there is however no convenient way to reuse the reconstructed image planes without uploading them to an SE somewhere; this will either leave the various reconstructed sub-groups spread across different SEs, or else again constrain the jobs to using a particular resource.

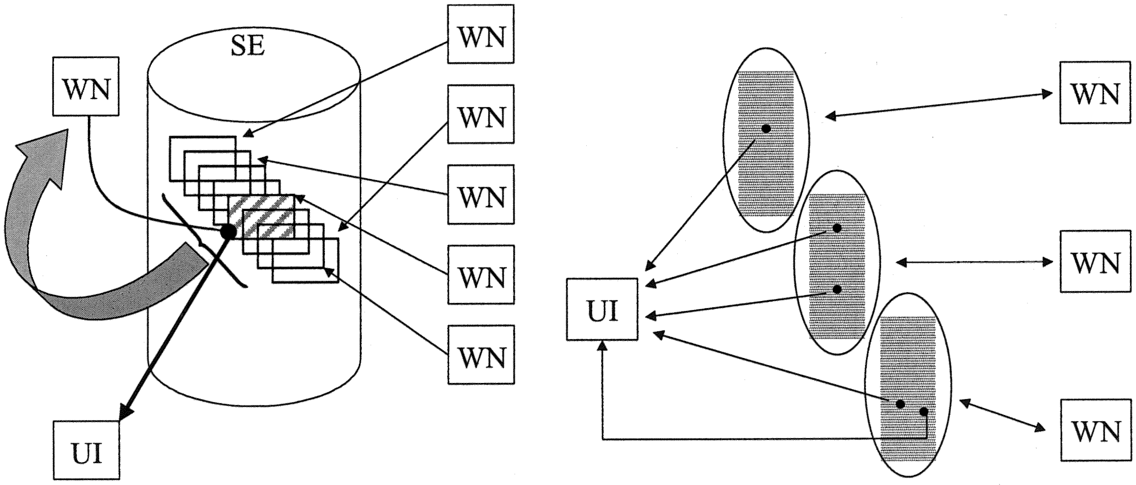

Figure 6: Object extraction with intermediate image storage on SE (*left*), reconstruction combined with extraction on WN (*right*)

In some applications of holography, for example studying predator-prey relationships in oceanic plankton[9], only a volume surrounding a key prey species (usually large compared to the photosynthetic food) is needed plus one control volume located as far away from a predator species as possible within the same overall volume. In such cases it would be computationally advantageous to run a pre-processing job to locate the region in (x,y) which contains the centre of the fringe pattern of the large prey object. With this information, and a rather coarse depth sampling (every one or two millimetres), one could quickly eliminate much of the volume from detailed reconstruction and processing. This coarse reconstruction would also provide a human with an overview of the sample volume, and thus the opportunity to identify regions of particular interest (or conversely that the entire volume contains nothing of current interest and need not be processed further). Obviously there are other applications of holography, such as Particle Image Velocimetry[28], where such an approach would not be appropriate.

5.3. FFTW and plan choice

The FFTW library used by the HoloPlay reconstruction code is not a single algorithm, but instead consists of a set of algorithms and strategies from which can be selected a particular "plan" that best suits a given combination of machine and problem[18]. Either a specific plan may be enforced by loading from a "wisdom" file, or more commonly the optimal plan is identified by FFTW at run-time by profiling the machine. For large images the profiling step is insignificant and the possible benefits make it worth doing for each job. For small, CCIR size, images however table 1 implies that the profiling time dominates. Since the WNs at a particular CE (which can be specified) are near-identical at worst and the program and data are the same, it may be worth investigating whether using a fixed plan gives better performance overall, if the gain from many optimised runs outweighs the loss from the occasional non-optimised ones. As FFTW is fairly widely used, it might also be worth considering having each WN provide its own wisdom information as part of the Grid infrastructure, rather than repeatedly re-calculating it at user level.

CONCLUSIONS

Improvements in sensors and image processing are making possible new applications for digital holography, but demand ever-larger sensors and faster frame-rates, and hence greater computing resources for volume reconstruction. We believe that Grid computing will provide an effective and convenient way to meet the computing needs of novel holographic instruments, and have proposed a number of deployment strategies that we are currently implementing.

ACKNOWLEDGEMENTS

The authors would like to thanks Marc Fournier-Carrié for his work with us in creating the original HoloBatch software. JJN and PRH would like to thank our HOLOMAR colleagues and the European Commission's MAST-III programme (MAS3-CT97-0079).

REFERENCES

1. M. Solan, J.D. Germano, D.C. Rhoads, C. Smith, E. Michaud, D. Parry, F. Wenzhofer, B. Kennedy, C. Henriques, E. Battle, D. Carey, L. Iocco, R. Valente, J. Watson and R. Rosenberg: "Towards a Greater Understanding of Pattern, Scale and Process in Marine Benthic Systems: A Picture is Worth a Thousand Worms" *Journal of Experimental Marine Biology and Ecology* **285** pp. 313-338 (2003)
2. C. Knox: "Holographic Microscopy as a Technique for Recording Dynamic Microscopic Subjects" *Science* **153** pp. 989-990 (1966)
3. C. Knox and R.E. Brooks: "Holographic Motion Picture Microscopy" *Proceedings of the Royal Society of London* **B174** pp. 115-121 (1969)
4. G.L. Stewart, J.R. Beers and C. Knox: "Application of Holographic Techniques to the Study of Marine Plankton in the Field and in the Laboratory" *Proceedings of the SPIE* **41** pp. 183-188 (1973)
5. D.K. Costello, K.L. Carder, P.R. Betzer and R.W. Young: "*In situ* Holographic Imaging of Settling Particles: Applications for Individual Particle Dynamics and Ocean Flux Measurements" *Deep-Sea Research* **36** (10) pp. 1595-1605 (1989)
6. J. Katz, P.L. Donaghay, J. Zhang, S. King and K. Russell: "Submersible Holocamera for Detection of Particle Characteristics and Motions in the Ocean" *Deep-Sea Research I* **46** pp. 1455-1481 (1999)

7. R.B. Owen and A.A. Zozulya: "In-line digital holographic sensor for monitoring and characterizing marine particulates" *Optical Engineering* **39** pp. 2187-2197 (2000)

8. J. Watson, S.J. Alexander, G. Craig, D.C. Hendry, P.R. Hobson, R.S. Lampitt, J-M. Marteau, H. Nareid, J.J. Nebrensky, M.A. Player, K. Saw and K. Tipping: "Development and sea trials of a subsea holographic camera for large volume *in situ* recording of marine organisms" in *Practical Holography XVI and Holographic Materials VIII,* Stephen A. Benton, Sylvia H. Stevenson and T. John Trout, Editors, Proceedings of SPIE vol. **4659** pp. 69-75 (2002)

9. P.R. Hobson and J. Watson: "The Principles and Practice of Holographic Recording of Plankton" *Journal of Optics A: Pure and Applied Optics* **4** pp. S34-S49 (2002)

10. G. Craig, S. Alexander, S. Anderson, D.C. Hendry, P.R. Hobson, R.S. Lampitt, B. Lucas-Leclin, H. Nareid, J.J. Nebrensky, M.A. Player, K. Saw, K. Tipping, J. Watson. "HoloCam: A subsea holographic camera for recording marine organisms and particles" in *Optical Diagnostics for Industrial Applications*, Neil A. Halliwell, Editor, Proceedings of SPIE vol. **4076** pp. 111-119 (2000)

11. J. Watson, S. Alexander, S. Anderson, G. Craig, D.C. Hendry, P.R. Hobson, R.S. Lampitt, B. Lucas-Leclin, H. Nareid, J.J. Nebrensky, M.A. Player, K. Saw and K. Tipping: "The Development, Construction and Test of a Subsea Holographic Camera ("HoloCam") for Recording Marine Organisms" Proceedings of *Oceanology International 2000*, 7th-10th March 2000, Brighton, UK, pp. 183-192 (2000)

12. J.J. Nebrensky, G. Craig, P.R. Hobson, R.S. Lampitt, H. Nareid, A. Pescetto, A.Trucco and J. Watson: "A data extraction system for underwater particle holography" in *Optical Diagnostics for Industrial Applications*, Neil A. Halliwell, Editor, Proceedings of SPIE vol. **4076** pp. 120-129 (2000)

13. J.J. Nebrensky, G. Craig, G.L. Foresti, S. Gentili, P.R. Hobson, H. Nareid, G.G. Pieroni and J. Watson: "A Particle Imaging and Analysis System for Underwater Holograms" Chapter 8 (pp. 79-91) in *Optical Methods for Data Processing in Heat and Fluid Flow*, Professional Engineering Publishers (2002)

14. H. Sun, H. Dong, M.A. Player, J. Watson, D.M. Paterson and R. Perkins: "In-Line Digital Video Holography for the Study of Erosion Processes in Sediments" *Measurement Science and Technology* **13** (10) pp. L7-L12 (2002)

15. Fairchild Imaging CCD595. On-line [available http://www.fairchildimaging.com/main/area_595.htm]

16. T.M. Kreis, M. Adams, and W.P.O. Jüptner: "Methods of Digital Holography: A Comparison" in *Optical Inspection and Micromeasurements II*" ed. C. Gorecki, Proceedings of SPIE vol. **3098** pp. 224-233 (1997)

17. M. Fournier-Carrié: "*Computer Processing in Digital Holography*" Department of Electronic and Computer Engineering, Brunel University, unpublished (2001)

18. FFTW. On-line [available http://www.fftw.org/]

19. Atmel Corporation. On-line [available http://www.atmel.com/dyn/resources/prod_documents/doc1986.pdf]

20. I. Foster and C. Kesselman: "*The Grid: Blueprint for a New Computing Infrastructure*" 2nd revised Ed. Morgan Kaufman (2003)

21. I. Foster, C. Kesselman and S. Tuecke: "The Anatomy of the Grid: Enabling Scalable Virtual Organizations" *International Journal of High Performance Computing Applications* **15** (3) pp. 200-222 (2001) [available online: http://www.globus.org/research/papers/anatomy.pdf]

22. The Globus Alliance. On-line [available http://www.globus.org/]

23. The EU DataGrid. On-line [available http://www.eu-datagrid.org/]

24. The LHC Computing Grid. On-line [available http://lcg.web.cern.ch/LCG/]

25. The LCG Grid Operations Centre. On-line [available http://goc.grid-support.ac.uk/gridsite/gocmain/monitoring/]

26. C. Grandi and A. Renzi: "Object Based System for Batch Job Submissions (BOSS)" *CMS Note 2003/005*, (2003); On-line [available http://www.bo.infn.it/cms/computing/BOSS/]

27. D. Bonacorsi, D. Colling, L. Field, S.M. Fisher, C. Grandi, P.R. Hobson, P. Kyberd, B. MacEvoy, J.J. Nebrensky, H. Tallini and S. Traylen: "Scalability Tests of R-GMA Based Grid Job Monitoring System for CMS Monte Carlo Data Production" *IEEE Transactions on Nuclear Science*, in press.

28. G. Pan and H. Meng: "Digital in-line holographic PIV for 3D particulate flow diagnostics" *4th International Symposium on Particle Image Velocimetry*, Gottingen, Germany (2001)